Operations Research
in Business Administration and Management

Concepción Maroto
Javier Alcaraz
Concepción Ginestar
Marina Segura

Operations Research
in business administration and management

Concepción Maroto Álvarez
Javier Álcaraz Soria
Concepción Ginestar Peiro
Marina Segura Maroto

EDITORIAL
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

http://tiny.cc/edUPV_rea

The publication of this book has been approved by the Department of Applied Statistics and Operations Research and Quality of the Universitat Politècnica de València (Spain)

Academic book collection

To cite this publication please use the following reference: MAROTO ÁLVAREZ, C., [et al] (2014) *Operations research in business administration and management*. Valencia: Universitat Politècnica de València

First edition, 2014

Authors: Concepción Maroto Álvarez
Javier Álcaraz Soria
Concepción Ginestar Peiro
Marina Segura Maroto

Cover design by Carmen Lloret

Of this edition: edUPV / www.lalibreria.upv.es / Ref.: 6176_02_01_01

ISBN: 978-84-9048-242-1 (print version)
ISBN: 978-84-9048-241-4 (electronic version)
DOI: https://doi.org/10.4995/REA.2022.617602

To our students: never lose the hope to do things better.

Acknowledgments

We acknowledge the support of the Faculty of Business Administration and Management of the Universitat Politècnica de Valencia in preparing this book for Operations Research teaching.

We also thank Stephen and Dogan for their infinite patience with the English proofreading.

CONTENTS

CHAPTER 1

THE NATURE AND METHODOLOGY OF OPERATIONS RESEARCH

This chapter begins with an overview of the origins and evolution of Operations Research in order to understand its current and future interest in Business Administration and Management studies. Due to the lack of a generally accepted definition for the discipline, several definitions are presented in section two of this chapter, in an attempt to cover the traditional approach as well as some more recent approaches. Section three describes some applications in Business Administration and Management. Lastly, the methodology of Operations Research is presented, including an in-depth analysis of the problem formulation, modelling and implementation.

1.1. THE ORIGIN AND EVOLUTION OF OPERATIONS RESEARCH

The term "Operations Research" or "Operational Research" (OR) was first used in the UK, when in 1936 a group of RAF scientists was established to study how to operate radar. This group was called the Operational Research Section, as it was more concerned with the operations of the new equipment than with the development of radar itself. This first section was so successful that by the time World War II was over other allied nations already had similar support groups for their military operations decision making (Keys, 1995). However, Kirby (2000) places the formal birth of Operations Research in 1938.

The positive impact of Operations Research during World War II brought about its subsequent smooth introduction into the industrial and commercial sectors on both sides of the Atlantic. In England it was concentrated in two industries, coal mining and the iron and steel industry. Paradoxically, the teaching of Operations Research started in the United States sooner than in England. The first textbook on the subject was written in the USA: Churchman, C. W., R.L. Ackoff and E.L. Arnoff (1957): *Introduction to Operations Research.* John Wiley, New York. In the 1960s American industry was therefore receptive to Operations Research techniques due to the preparation of its managers (Kirby, 2000).

It should be noted that these origins of Operations Research represent the official history accepted by the majority. Other authors such as Professor Bueno (1971) expand on this and argue that there are three basic sources that influenced the appearance of Operations Research: economic models, military operations and mathematics. Thus, he finds evidence in, amongst others, Walras's general balance model, which outlines the production balance with a system of linear equations, the Tableau Économique of Quesnay (1758) and in Input-Output Analysis by Leontiev (1936). However, the special relevance of Operations Research to 20th century economic thinking becomes evident as many Nobel Prizes of Economy (which started in 1969) have been awarded to authors who published works in the area of linear programming and other quantitative techniques and nonlinear programming and games theory. Among these Nobel prizewinners are Samuelson (1970), Leontiev (1973), Kantorovich (1975), Simon (1978), Solow (1987), Markowich and Sharpe (1990), Selten, Nash and Harsanyi (1994), Aumann (2005) and Hurwicz (2007).

Keys (1995) considers the period 1945-1975 as a period of growth and stability that he calls the “Golden Age” of classic Operations Research. During this period the institutionalization of Operations Research is seen in three aspects: as a support to management in industry and other organizations, in the creation of professional associations and in the establishment of academic programs.

The historical work carried out by Kirby (2000) emphasizes the rapid dissemination of Operations Research techniques after the revolution of information and communications technologies during the 1980’s. Thus, in the early 1990’s, spreadsheets implemented linear and nonlinear programming codes, and their performance has improved a lot since then. This situation of Operations Research offers new challenges and opportunities for business administration and management. For example, nowadays millions of Microsoft Excel users can create and solve models that allow them to improve business decision making at operative, tactical and strategic levels. Moreover, the Operations Research techniques are and will continue to be part of the applications for helping decision making techniques known as *Decision Support Systems (DSS)* and *expert systems.*

1.2. THE NATURE OF OPERATIONS RESEARCH

From its beginning, there has not been a generally accepted and precise definition of Operations Research and, throughout its evolution, its methodology has been extensively discussed. The main characteristics of Operations Research appeared in the first OR textbook published in 1957. Specifically, there is an emphasis on the scientific method, the interdisciplinary teams, the decision making, obtaining the best solution and the global approach.

For Assad, Wasil and Lilien (1992) Operations Research - Operations Research/ Management Science (OR/MS) - is the application of the scientific method to decision making or to professions that approach the best way to design and operate systems, usually under conditions where allocation of scarce resources is required.

Keys (1995) considers Operations Research as a technology. Operations Research uses the scientific method on which its observation, modeling, thinking, experimentation and logical and systematic investigation are based. However, it does not use these methods with the same purpose as science. Science is descriptive, Operations Research is prescriptive. The objective of Operations Research is to provide information and to design ways to improve the effectiveness of organizations.

The two basic characteristics of this **technology** seen from this new model are their **objectives** and their methods. The objective of Operations Research is the **production of information** about the systems that facilitate improvement in the effectiveness of the organization. The **methods** used to produce this information are of a **scientific nature.** That is to say, they are based on measures, analysis and validation rather than likes, intuitions and opinions. If we consider that technology is in charge of designing systems, physical as well as abstract, Operations Research **is a technology that designs abstract systems** that consist of useful information for the planning, the control and the other necessary activities to manage an organization.

According to Keys, **Operations Research is a technology that designs abstract systems, by scientific means, to improve the effectiveness of organizations**. The implications for teaching and learning Operations Research take into consideration two components. On the one hand, formal means must be used to teach useful working ways, such as quantitative analysis and application of scientific methods. On the other hand, it is necessary to complement this education with the application of the previous skills to real problems. This textbook considers this approach as the most appropriate in areas where Operations Research is taught, which are, Business Administration and Management studies.

Robinson (2000) defines Operations Research as the application of the scientific method to improve the effectiveness of operations, decisions and management. Robinson considers that one of the reasons for the discipline to remain invisible or visible but not well understood, is because it has been practiced under different names. Besides *Operations Research*, other almost synonymous terms have been used such as *Management Science, Decision Technology, Decision Support, Policy Science, Systems Analysis* (with relative applications to administration and decisions), *Management Technology* and *Management Analytics.* ***Business Analytics*** is another recent name that integrates descriptive and prescriptive analytic methodologies.

An important feature of Operations Research is maintaining a **global perspective** on the projects, analyzing the particular problems in the context in which they occur. In both the classical and the most modern definitions of Operations Research the **system concept** is fundamental. Let us see some examples that will illustrate this.

Many companies calculate the unit production cost at a machine shop or a production line, taking into account all of the costs of the resources used. The lower the unit production cost the greater the efficiency. This procedure is valid only for production processes that consist of a single phase and where there is no trouble in selling the product. When the company has complex production processes with several products (e.g. tile companies) and each line produces different parts - often in small batches-, which are used as inputs in subsequent phases of the production process, this acts as an incentive for the machines to be producing all the time. If the next production line in the manufacturing process does not require these intermediate stocks immediately, the company will need to store them temporarily, incurring a cost for these intermediate stocks that are not attributed to the line that generated them. Therefore, the production line seems to be efficient, while the company has to deal with excessive costs caused by these intermediate stocks.

As we have seen in the previous example, the operation efficiency of a particular division of a company can impair the overall performance in terms of objectives and goals. The efficiency measures how well resources in a given activity are used. Thus we can speak of ***technical efficiency***, which does not need to be the same as ***economic efficiency***, which maximizes the difference between revenues and costs. However, a company is interested in achieving its objectives, which we can evaluate via its ***effectiveness***. That the several parts of a system operate efficiently does not necessarily mean that the whole system is effective in achieving its objectives. This is not to say that efficiency is contrary to effectiveness. **Real efficiency is measured in terms of the overall objectives of the company**. Efficiency and effectiveness are complementary concepts. In short, we can say that *effectiveness* deals with *"doing the right thing"* and *efficiency* with *"doing things right"* (Daellenbach and McNickle, 2012).

Here is another example to illustrate the concept of system in Operations Research. In a company with five departments (raw materials procurement, production, marketing, finance and personnel) marketing proposes an increase in the duration of the guarantee of one of its products to better compete. What forms the system? What forms the environment?

The marketing department consists of distribution, sales and customer services. The company assumes that extending the guarantee period will increase sales. However, they also increase guarantee costs due to added customer services. Therefore, the system to be studied could be reduced to sales and customer services (System 1), with all other operations of the company, customers and competitors which form the environment. The aim of System 1 is to find out the guarantee period that maximizes the difference between benefits from sales and guarantee costs.

System 1 considers product quality as a part of the environment, but product quality will affect both sales and warranty costs. For this reason, System 1 could be expanded to include production (System 2). The objective of this system is to determine the optimal combination of product quality and the guarantee period to maximize profits. However, product quality is also affected by the quality of the raw materials used, which are part of the system 2 environment. Thus System 2 could be expanded further to include the procurement of raw materials and form System 3. System 3 could also be extended to include other company's products, if sales of these products are affected by changes in the guarantee period of the first product, leading to System 4.

In Figure 1.1 we can see how each system is included in a larger one. With this example we want to illustrate the fact that the Operations Research always tries to solve the conflicts of interest within the company, so that the best result for the company in terms of its objectives is achieved. This does not mean that the study should always explicitly consider all aspects, but that the objectives sought after have to be consistent with those of the company.

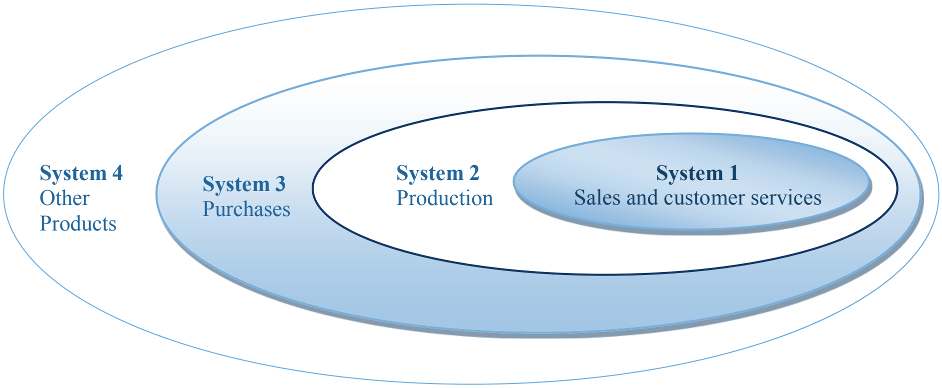


Figure 1.1. Example to ilustrate the system concept. Source: Daellenbach et al. 1987.

1.3. APLICATIONS

After World War II the British as well as the American army maintained active Operations Research teams. As a result, nowadays there is a large number of people called “military operations researchers” that apply the Operations Research approach to national defense problems. Operations Research is also widely used in other types of organization and in the business world. In fact, almost all large and many medium sized enterprises worldwide have established Operations Research teams.

Among the industries that apply Operations Research are those dealing with aviation and missiles, computer science, electric power generation, electronics, food, metallurgy, mining, paper, petroleum, transportation, as well as financial institutions, government agencies and hospitals. The companies which were finalists of the Franz Edelman INFORMS (***Institute for Operations Research and the Management Science***) prize provide excellent examples of real applications of Operations Research (http://www.informs.org).

Among the finalists in 2012 were Hewlett-Packard (HP), Intel and **TNT Express**. The latter won the award for its "Global Optimization", which uses advanced methods to optimize the transport network of the company. This program solves problems in warehouse location, optimal routes for trucks, fleet management and personnel scheduling.

In 2011 *Midwest Independent Transmission System Operator (Midwest ISO)*, a non-profit organization that **manages the electricity market** in 13 U.S. states (North Central region) and one in Canada (Manitoba), won the *award*. It has operational control of more than 1,500 energy production plants and 55,000 miles of power lines. It notifies the plants, every 5 minutes, of the amount of energy required to meet the current demand. It uses a linear programming model to calculate production levels and establish the market price of electricity. The model size is up to 3 million continuous variables and 4 million constraints. The solution to the model provides plant production levels and energy prices (shadow prices or opportunity costs). It also uses an integer programming model to determine when a plant should be producing or not. The individual companies retain physical control of the plants and transmission lines. *Midwest ISO* manages the real-time power to bid and buy on demand, manages the market and maximizes the benefit of the company which sells the cheapest electricity. In short, using techniques discussed in this book, in this example the price at which electricity is bought and sold is determined and, what is more important, the electricity is available when and where it is needed and provided safely.

In **2009 HP** and the **Marriot hotel chain** were notable entries with two Operations Research tools to manage the product portfolio and with a price optimizer respectively. Also in 2009 Zara was among the finalists for applying Operations Research to improve its distribution process. **Cocacola Enterprises**, the world's largest bottler and distributor of Coke products (Coke, Fanta, Sprite, Minute Maid, etc) was also recognized in 2007 for its application to schedule the daily routes of 10,000 trucks.

We will discuss the application of Zara in a little more detail. Zara´s supply chain consists of two main warehouses located in Spain, which regularly receive shipments of finished garments from suppliers and replenish all Zara stores twice a week. The key is to determine the exact number of each size (up to 8 different sizes) and each item (up to 3,000 at a time) to be included in each shipment to each store (there are over 1,500). Until 2005 Zara used a procedure that required a large number of employees to determine shipments to each store. The company developed a decision making process based on Operations Research methods, including methods of forecasting and a very large mixed **integer programming** model. The implementation of this new process presented many technical difficulties. One of them was to include the uncertainty of estimates and inventory policies of the stores, and the integration of a complex mathematical model with many large databases. They also had to have the software and hardware infrastructure necessary to solve optimization of thousands of problems in a couple of hours each day. Additionally, it presented challenges related to human resources, because the Zara corporate culture highly values intuition and personal judgment in decision-making. The development of this new process, supported by Operations Research techniques, was completed in all stores and articles and it has been used since 2007.

In general, **linear programming** and **integer programming** have been successfully used in solving problems related to the allocation of the means of production, material mixing, distribution, transportation, investment selection and planning of agriculture, among others. A very important application of linear programming in the field of economics is Data Envelopment Analysis (DEA), developed by Charnes, Cooper and Rhodes (1978). DEA is a linear programming based technique that allows us to empirically measure the productive efficiency of decision units such as groups of companies in the same sector, financial institutions, hospitals, educational institutions, etc. and identify the companies that are on the *efficient frontier* of production. The efficiency is measured by the weighted sum of the outputs on the inputs. The weighting structure is calculated using linear programming. Furthermore, the concepts of linear programming guide and facilitate the analysis and interpretation of the results of the DEA models. At present this is still a very active field of work, both in the application and in the research being conducted.

Nonlinear programming is also used in certain problems of resource allocation, selection of efficient portfolios, new product design, production problems, mixtures in chemical processes, etc. **Multiobjective programming** and **goal programming** also have many applications such as natural resource management (Weintraub et al., 2007), scheduling of advertising media, land use management, location of utilities and planning of resources in hospitals to name just a few. Other Operations Research techniques such as inventory theory, game theory and simulation have been used in a variety of contexts.

Operations Research shares with Artificial Intelligence the objective of providing methods and procedures for solving problems and making decisions. Artificial Intelligence is inferential and has expert knowledge and heuristic methods. Operations Research is mainly based on mathematical algorithms. A careful integration of these two approaches has a bright future ahead for the performance and acceptance of the systems. **Decision Support Systems** for decision making integrate Operations Research and Artificial Intelligence techniques in information systems that are very useful in the decision making process. This integration can make Operations Research techniques more accessible to decision makers and the models can also use Artificial Intelligence techniques. We should also highlight heuristic search techniques such as genetic algorithms, tabu search and simulated annealing.

Operations Research models are common in **finance**, often grouped under the name of **financial engineering**. Similarly **marketing engineering** usually means Operations Research applied to marketing. In this field it is applied to strategic decisions (planning, portfolio, etc.) and at the tactical level (product design, advertising, etc.). They also play an important role in the analysis of electronic markets. Other opportunities will come from electronic trade and investment, from online banking to online insurance.

With regard to supply chain management, the digital economy provides opportunities to use Operations Research in resource planning in companies. Given the information that is available online, advanced planning and production scheduling, will improve coordination and cooperation between suppliers and customers. The growth of mobile computing and communication will increase the aid that applications give to decision-making in transport trucks. Thus there are companies that optimize loading and truck routes using web applications to obtain data and distribute solutions. The Internet also facilitates the expansion of supply management towards integrating product design, sales and customers.

Finally, we must emphasize the strengths of **Operations Research** in the digital economy era: it exploits the vast amount of data available, which is of ever increasing complexity due to its analytical nature and uncertainty, modeling **increases our understanding of business processes and virtual experiments can be made without risk to business and thus provides decision making technology** for the automation of recurring decisions in real time, such as for web applications. In short, the Operations Research of the future is Operations Research in real time. Customers often ask when their order will be delivered. Providers base their response on their inventory and the scheduled production in progress. However, they should now be able to respond after performing a scheduling algorithm including the potential order. To achieve the required performance in real time sometimes we need to resort to heuristic algorithms such as those discussed in the last chapter of the book.

1.4. METHODOLOGY OF OPERATIONS RESEARCH

Daellenbach and McNickle (2012) clearly establish three major phases in the methodology of Operational Research which are: problem formulation, modelling and implementation, which in turn break down into the sub-phases indicated in Figure 1.2.

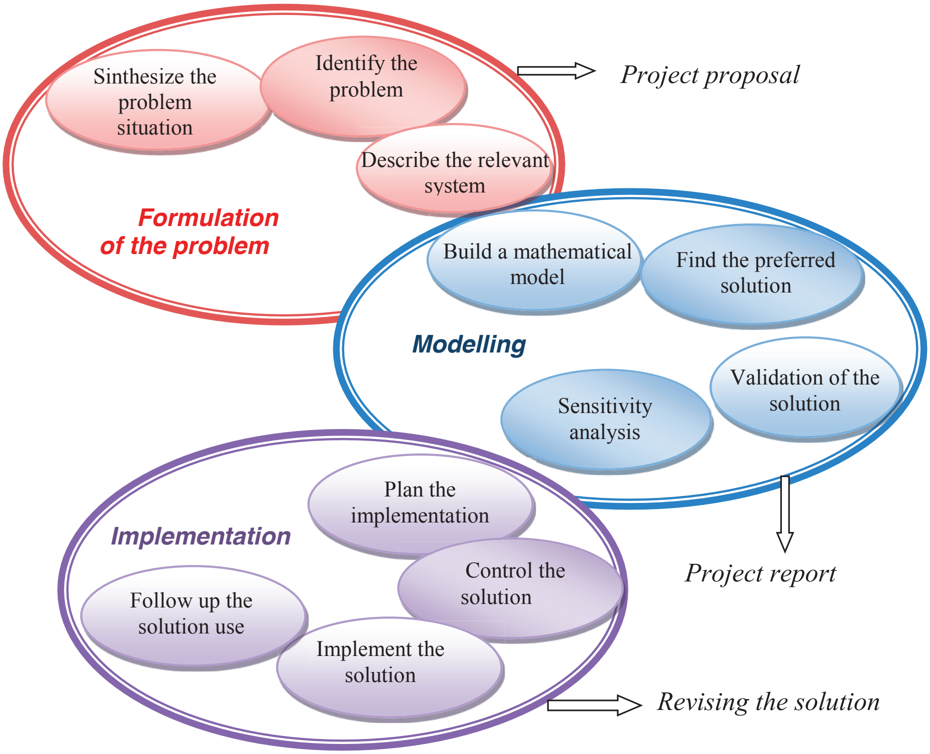


Figure 1.2. Methodology of Operation Research. Source: Daellenbach and McNickle (2012)

1.4.1. FORMULATION OF THE PROBLEM

In the first place, we should make a **synthesis of the situation**, for example through **graphics or charts** which will help us during the problem definition phase, after which we should identify the structure, the transformation processes, the components and the inputs and outputs of the relevant system. For a problem to exist there must be an individual or group of individuals, called **decision-makers**, who are not satisfied with the current situation or who have unsatisfied necessities, such as reaching some goals or objectives. They also know when the goals or objectives have been satisfactorily reached and they have control over the aspects of the situation that affect the extent to which the goals or objectives are achieved. The four elements of a problem are:

- The decision-maker/s.
- The decision-maker´s **objectives**.
- The **measurement** of efficiency in order to be able to assess the extent to which **objectives** are achieved.
- The **action alternatives** or **decision variables** to reach the objectives.

The second step of the formulation of the problem is its identification and consists of defining these four elements. The third step consists of defining the relevant system for the problem that we have identified in the previous step, including its environment. The decision-maker has an esential role in the problem formulation phase.

In practice, the determination of these four components might not be so easy to obtain by simply asking the decision-maker. Sometimes the decision-maker only has a vague intuition that things could go better. We should explore and clarify the situation through several people involved in the situation. Sometimes, it may happen that the person who makes the decisions does not have access to the information needed to make an effective decision and the one that has the information does not have enough authority to make decisions. In these cases, the first thing to do is to change the structure of the organization, re-assigning the roles in the decision making. In most real applications, problem formulation is not achieved in these three steps, rather the initial formulation is detailed with successive reformulations, as the problem is better understood. In fact, it continues until the project concludes. However, it is in this phase where the success or failure of many projects occur.

Once we know the problem and the relevant system well enough, we can decide whether Operations Research may provide a solution to the problem. Therefore we should ask ourselves the following questions:

Can the problem be expressed in quantitative terms?

Are the required data available or can they be obtained at a reasonable cost?

Does the cost of the analysis justify the possible benefits that will be obtained from the implementation of the results? To what extent can the decision-maker expectations be fulfilled?

If we answer these questions affirmatively, then the formulation phase concludes with a proposal that will be the document which the decision-maker will use to decide to continue with the project or not. Therefore, the proposal is a key element. We should not promise more than we know that we can obtain with the available resources. Since Operations Research has much in common with the scientific research, it should be guided by the **ethics** of the scientific method.

The following anecdote from Ackoff illustrates both the difficulty of formulating the problem in real cases, and the fact that we can not always solve an unsatisfactory situation by making models. It is as important to know what models are useful and when we can improve decision making, as it is to know how to recognize when they are not the right tool. The administration of a large office building received complaints for years about excessive staff time spent waiting for the elevators in the main lobby. Several teams of Operations Research analyzed this problem of excessive waiting time. Different solutions were proposed: to use some elevators for lower floors and others for higher floors only. However, it was concluded that a significant reduction would be possible only by installing new elevators with a high associated cost. A member of the last team to study the problem asked why staff complained and after appropiate inquiries it turned out to be because of boredom. The Operations Research team then proposed installing mirrors. Some workers used them to make a final check of themselves or to check out other staff without being too obvious. When this solution was implemented the complaints disappeared (Ackoff, 1987). Nowadays, screens with information of interest to the staff can achieve the same effect as the mirrors did then.

Operations Research, in many cases, does not intend to find the optimal solution, but to find some degree of improvement over the previous situation. One of the founding fathers of the discipline colloquially explained it as follows: "Operations Research is the art of providing bad solutions to problems that otherwise would have worse solutions."

1.4.2. MODELLING

This phase distinguishes Operations Research from other methods of solving problems. According to Daellenbach and McNickle (2012), Operations Research is often seen as a number of techniques and mathematical tools, which do not favour the discipline at all to the detriment of its potential. The modelling phase begins by expressing the system related to the problem in quantitative terms. A mathematical model expresses in quantitative terms the relationships between especially important components of the system that have been defined in the formulation phase. These relationships can sometimes be represented in a spreadsheet and for some others it is necessary to formulate the relationships in terms of mathematical expressions, such as equations, inequalities or functions. The term model is used in a broad sense, since it can take the form of a chart as well as of mathematical expressions.

We call the action alternatives or controllable aspects of the problem **decision variables**. The term **action alternatives** is used when the number is discrete and usually small. The measurement of the behaviour or effectiveness is the aspect that measures the extent to which the objectives of the company are reached. If this measure of effectiveness can be expressed as a function of the variables, we call it the **objective function.** Our goal is to find the values of the decision variables that maximize or minimize the objective function. The **parameters or coefficients** represent the uncontrollable aspects of the problem. And the **constraints** are the mathematical expressions that limit the range of values of the decision variables. From the early 50's, a number of mathematical models have been developed with their own resolution procedure, such as linear programming and its numerous extensions, network models such as the critical path, etc. They are what we call general purpose models. Any problem that satisfies the hypotheses of a general model can be approached and solved in this way. For those problems that do not adapt to any specific technique of Operations Research, a model should be developed for the specific purposes, with a unique structure for that particular problem. Likewise, a solution procedure has to be created for that specific case. Lastly, when all of the inputs and relationships are known, the problem is deterministic, while if some inputs or results are subjected to uncertainty, like the probabilistic influences, the model is known as probabilistic or stochastic.

A mathematical **model,** to be useful, should enable **better decisions** when you use it than when you don't and should also be:

- **Simple:** Simple models are easier for the decision-maker to understand. It will be easier if the decision-maker follows the logic of a spreadsheet than a set of equations. However, when designing complex models it is unavoidable that approximations appropriate to the real situation be made.
- **Complete:** The model should include all of the significant aspects of the problem that affect the measurement of its effectiveness. It may be necessary to design two models, one with certain aspects to compare and to decide their relevance and another one without them.
- **Easy:** It should be possible to obtain responses from the model, such as the optimum solution, with a reasonable computational effort. Moreover, it should be easy to prepare, update and change the parameters and obtain new answers quickly.
- **Adaptive:** Usually, reasonable changes in the structure of the problem do not invalidate the model. If the changes invalidate the model, it may be possible to adapt the model with slight modifications. An adaptive model is known as a **robust** model.

In practice, we may find these properties useful in a model, however, users may only appreciate some of them. Thus, the decision-maker and the user of the model might be more interested in the desirable properties of the **modelling process** than in those of the model itself. The **credibility** and **trust** of the user are related more to the process and the interaction with the modeller than with the model itself. In this sense, it is important to keep in mind the following aspects:

The model should be appropriate for the situation under study: The model produces outstanding results with the smallest possible cost and in the time required by the decision-maker. A “good” Operations Research model does not necessarily have to show the details or to resemble the physical system that it attempts to optimize. In addition, a good model should allow us to measure the progress reached toward the decision-maker’s objectives.

The model has to produce information that is relevant and appropriate to the decision-making process. If the model complies with these last two properties and we can demonstrate them to both the decision-maker and the user, then it is more likely that they will find the model **useful**. Lastly, some of the properties of good models are in conflict. A simple model cannot take into consideration all of the relevant aspects. A robust model cannot be simple. A model that includes all of the significant aspects may not be easy to manipulate. The person (or team) that builds the model should balance these aspects and adopt a commitment, which should take into consideration the funding and the time available for the analysis. It should also take into consideration the possible benefits. Thus, the use of simple and quick strategies that provide 50% of the benefits can be economically more advantageous than using a sophisticated and expensive model that achieves 90% of the potential benefits. The cost of development of the mathematical model, data collection, calculation of the best solution, model implementation and maintenance, increases more than proportionally with the sophistication of the model, while the benefits increase less than proportionally.

Although the **process of mathematical modelling** can be considered as a scientific process, there are **certain aspects that are closer to an art than to a science**. It is considered an art because it is necessary to develop simple models that are good approaches to real life. There is little advice that can be given in this respect, except that the ability and necessary skills can be acquired with practice. Experts recommend starting with simple models that become richer evolving towards elaborated models by the incorporation of additional aspects of the problem. Another tip is to work with numeric examples, as well as graphics and charts.

The construction of models in the practice of business administration and management is valuable at least for the following reasons (Eppen et al, 1997):

1. The models require that the objectives be explicity defined.

2. The models require the identification and recording of the types of decisions that influence these objectives.

3. The models require the identification and recording of the interactions between all these decisions and of their respective advantages and disadvantages.

4. The models require thinking about the variables to include and to define them in quantitative terms.

5. The models require us to consider what data are relevant for the quantification of these variables and to determine the interactions between them.

6. The models require the recognition of the relevant restrictions on the values that variables can take.

7. The models can communicate ideas and expertise, facilitating teamwork.

After the construction of the model, we manipulate the quantitative model to explore the system's behaviour in response to the changes in the inputs, that is, we explore the solution space. The objective is to find the preferred solution in terms of the decision-maker's objectives. If the latter is interested in a main objective the optimal solution has to be found. Thus, if the measurement of the efficiency is the benefit, the optimal solution is the one which maximizes the benefit.With the validation and evaluation of the solution, the credibility of the model is then established, in the sense of being a valid representation of the reality. We should ask ourselves what improvement, in terms of benefit or cost savings, and what range of potential benefits can be expected. The answers to these questions will determine whether we should abandon the project, readdress it or continue in the same line.

In summary, we can conclude that the solution of the model can be found by enumeration if the number of alternatives is relatively small (dozens), with search methods like those of interval elimination, the algorithm-based methods, classic calculus methods, heuristic solution methods and simulation. The most powerful solution methods are those based on **algorithms.** An algorithm is a group of logical and mathematical operations that repeat in a certain sequence. Each repetition of these rules is called an **iteration.** An algorithm begins with a solution that improves at each iteration whose solution becomes the initial solution of the following iteration. This process repeats until the stop rule is met. This may indicate that the optimal solution has been reached or when the number of iterations or established computing time has been reached.

For an algorithm to be a practical method of problem solving it should have several properties:

- Each successive solution must improve the previous one.
- The successive solutions must converge, being closer and closer to the optimal solution.
- The convergence on the optimal solution must happen within a reasonable number of iterations.
- The computational effort of each iteration has to be sufficiently small in order to be economically acceptable.

The upper bound of iterations depends on the potential benefits of the problem. What can be reasonable to find the optimal strategy for expansion in a large company, may be excessive to find the optimal route for delivery in a city. A computer is needed in order to make these calculations in real problems. Many general Operations Research techniques such as linear programming or integer programming use algorithms to find the optimal solution.

When the models are highly complex or impossible to handle with the computer equipment available, heuristic methods may be the only possible alternative, as discussed in the last chapter of the book (genetic algorithms and other metaheuristics). Heuristic methods use ingenuity, creativity, intuition, knowledge or human experience to find optimal solutions or to improve an existing one. Sometimes, the explanation of why some rules lead to good and even optimal solutions can be found. In other occasions, they are used because they have proved that they work well. In general, **heuristic methods** do not guarantee the optimal solution, although they may find it in some cases. According to Herbert Simon's terminology, we could say that heuristic methods are usually more associated with the word "*satisfy*" than with optimize. Lastly, the simulation model is used in complex dynamic systems, especially in those where it is necessary to take into consideration stochastic aspects. In this case, as in the heuristic methods, the analyst should expect to obtain good policies, rather than the optimal one.

The last two steps of the modelling phase consist of the validation of the model, the evaluation of the solution and carrying out the sensitivity analysis. To be more exact, the objective of model validation is to establish whether the model is mathematically correct, logically robust and describes the reality well enough. Model validation has two facets, the internal validation known as **verification** and the external validation that is called **validation.** Verification involves analyzing whether the model is mathematically correct and logically robust. The best method to verify a model is to analyze the results obtained by testing the model using numerical data with a wide range of values. This involves verifying whether the expressions are dimensionally homogeneous. We should verify the correction of the numerical constants. We should also verify the model during its development, in order to find out the possible overlapping mentioned above between the different phases of the methodology. In complex models with many interrelated expressions the logical consistency should be verified.

The external validation is much more complex than the verification. Whether the model is or not an adequate approximation of the actual situation is a matter of judgement. This will depend on the purpose of the model and the use of the solution. One approach will be enough for an exploratory, while a model used to make decisions daily will have to reflect the reality better. Therefore, validation is a phase that overlaps with the definition of the relevant system and with the construction of the model. We should keep in mind that the validity of the model cannot be proven, but only seen as non valid. This has to do with the credibility of the model. If the model is credible, the user will trust it. Therefore, the decision-maker and the user play an important role in the external validation. Model verification and validation are often seen as phases that are carried out when the modelling phase has been completed. This is an erroneous position. If it is done at this time, errors or questionable hypothesis will waste time and effort. The evaluation of all aspects of the model is a continuous process.

With regard to the evaluation of the solution, the main objective is to determine the prospective benefits, such as benefits or net savings obtained by implementing the solution. Model evaluation is usually carried out by means of computer simulations, by comparing the current situation and the proposed one with the same data set. If the system is not yet in operation, it can be evaluated by estimating the potential benefits from artificially generated data. Among the rules for the evaluation of the solution are:

- The evaluation of the proposed solution should be based on observations of the real (simulated) behaviour within a sufficient period of time.
- Data should be independent of the data used in order to obtain the best policy.
- The trials should not only provide the expected behaviour, but also some measurement of their variability.

- Lastly, we would like to emphasize the similarity between the Operations Research projects and the research and development projects of new products. It is necessary to invest a certain amount of funds before it is possible to know the potential success of the project.

The last step of modelling consists of carrying out what is called "*What-if*" analysis. How do the individual or simultaneous changes of the non controllable inputs of the system affect the selected solution or the optimal solution? How much does the use of the model with incorrect parameters cost in terms of reduction of benefits? The sensitivity analysis allows us to answer these questions and it is, without doubt, one of the most important steps in the Operations Research method. The knowledge of the problem from sensitivity analysis may be more valuable than the solution itself. In short, **sensitivity analysis** consists of the **systematic evaluation of the optimal solution response to the modifications of the input data**. This analysis allows us:

- To determine the **necessary accuracy** of the input **data** for the model.
- To establish **control intervals** for modifications in the **parameters** and input **data** for which the optimal solution remains almost optimal.
- To evaluate the **dual price** of **scarce resources**.

It is known that some techniques, especially linear programming, provide a certain quantity of sensitivity analysis, either as a by-product of the algorithmic calculations or with little additional effort. In other cases, the sensitivity analysis requires that the problem be solved for several combinations of the input data.

When the modelling phase ends we prepare a detailed report of the analysis carried out, including achievements and recommendations for its implementation. This document is as important as the proposal, since the implementation will be decided on this basis.

1.4.3. IMPLEMENTATION

In the implementation phase, we should first prepare a detailed plan of the different tasks, who they should be assigned to and how they should be coordinated. Next, the procedures for the establishment and maintenance of controls for the recommended solution are proposed. For example, to specify the range of values of the model parameters for which the current solution remains valid and the exact updating procedure when they exceed these limits.

In the implementation of the solution the required changes are made in order to move from the current situation to that of the proposal. Part of this step comprises the preparation of complete documentation for the model, the software developed for its use and the corresponding user manuals. Finally, after the solution has been checked during a reasonable period of time we should revise it again. In this phase the extent to which the solution satisfies the expectations in terms of benefits achieved and costs incurred must be verified, as well as a check on whether it is being used as it should be used and whether there are recommendations for improving the model and the software which have arisen out of the use of the system and the experience acquired. This step may lead to a final report.

The **implementation process** is full of **difficulties** that are due to **human nature**. This is especially the case in projects that attempt to improve existing operations. Implementation problems may come from three sources:

Firstly, the problems related to the implementation of physical tasks will be the complexity of the solution, the sensitivity of the benefits or the costs to deviations from the rules set forth and the extent to which the proposed solution deviates from current solutions. Secondly, problems related to users and other individuals affected by the solution, such as their personalities, their motivation and their pride in their work. Thirdly, problems related to the project environment, such as support for the project and its solution by higher authorities (where management support is not visible and explicit, the users will not collaborate and support the project as much as when management is clearly behind the project) and the organizational implications of the solution (if the user's department is more dependent on another or if the users perceive that the solution threatens the security of their positions they will provide less support).

We generally concentrate more on the first factor, which is a matter of technology, devoid of human aspects. The second and third factors are of a qualitative nature and they are usually neglected and overlooked. However, they can be serious restrictions to the model implementation. From this point of view, we can see **implementation** as a **problem of relaxing human restrictions versus the adjustments of the technical solution.** A way to relax these restrictions is to further involve the user from the beginning of the project and train them so that they understand and feel themselves to be a part of the solution. The success of the implementation may be guaranteed if the **decision-maker** and the **user** feel that they are the owners of the results of the analysis. They can develop this feeling of ownership if they can contribute to the project in different ways, with their experience and deep knowledge of the operations. Therefore, the analyst should keep them informed and ask for their opinion and advice regarding many ideas. If they feel that they have **contributed significantly to the project**, they will want to see the solution in operation and they will have an active role in the implementation process.

We have already said that the planning of model implementation should begin with the start of the project, when the first contact with the sponsor is established. A general guide to planning the implementation is the following:

- To identify all those stakeholders involved in the problem, in particular, decision-makers and users. The former because they have to approve the model implementation and the latter because their cooperation is necessary for the continued use of the solution.

- To establish effective lines of communication with the decision-makers and users.
- To explore and to manage the previous expectations of the project.
- To keep owners and users regularly informed.
- To check the availability and sources of all the necessary data.
- To order the required equipment and software, if necessary.
- To develop all the necessary software for the implementation and use of the new solution.
- To plan and perform the real process of implementation, such as the preparation of all data in the required form, user manuals, training sessions, etc.
- To hold regular monitoring sessions with users.

Project management techniques may help us to carry out the planning of the implementation appropriately. With regard to the control and maintenance phase of the solution, the following should be done:

- For each parameter and constraint we should indicate the quantitative changes for which the solution remains optimal or almost optimal.
- Point out the structural relationships between inputs, variables and results that are assumed by the model.
- Specify how to measure the inputs and with what frequency to see if the changes are significant and, if so, what measures should be taken.
- Assign the responsibilities for the control of each item and identify the person to be notified about the detected changes.
- Specify how to adjust the solution in response to quantitative changes in the inputs and who is responsible for these changes, who else should be informed and what actions should be taken in case of possible structural changes.

The work is not finished with the model implementation, as it is essential to monitor the behavior of the model for some time. If there is a misunderstanding or incorrect application, measures must be taken. For example, correcting the manuals, scheduling training sessions, etc. One of the last things to be done is to audit the solution. This involves evaluating the extent to which the expected benefits of the project have been obtained. For the comparison between the benefits before and after the project to be valid, as we have already stated previously, we should use the same data. If important discrepancies are found between the real benefits and those expected in the project we should analyze and explain the causes. This is not only important for the sponsor of the project, but also for the analyst, who can obtain a quantitative and qualitative feedback from his own performance. Lastly, as Daellenbach and McNickle (2012) point out, the implementation of all of the recommendations of an Operations Research project is not usual. Therefore, the analyst's goal should be to achieve a sufficiently high level of model implementation so as to be able to obtain most of its potential benefits.

As seen in the previous section, **Operations Research attempts to improve the effectiveness of the system at a global level**. This is only possible if the solution is implemented as completely as possible. **Assuring the implementation of the solution** is one of the first things to consider during the formulation and modelling phases. All measures that increase the possibilities of a complete implementation should be planned from and start with the beginning of the project and continue throughout all phases. Although we have already mentioned it, we would like to emphasize that the natural order in which **the phases of an Operations Research project** have been described is the order in which they usually start, but that each step **overlaps with the previous and subsequent steps**. As for the selection of the most appropriate model, the cost of the model development as well as model implementation should be considered. For example, if the required personnel qualifications are higher than those currently available, it might be preferable to choose a simpler model. We would certainly obtain less benefit, but greater possibilities of implementation.

The methodology of Operations Research is also iterative, which implies that the analyst may have to revise previous steps and redo or modify analysis already carried out. We sometimes discover during the process of resolution of the model that it is very expensive in terms of computing. We should then return to the model construction phase and build another model with fewer requirements. It is also possible that during the model implementation phase we notice serious errors that mean we need to start again with a new model formulation, provided that the decision-maker agrees. According to Daellenbach and McNickle (2012), it is important to have complete documentation for the project, thus it is advisable that the analyst keeps a project log in which the hypotheses and simplifications which have been taken into consideration are recorded for further analysis and validation. It is easy to overlook or forget these decisions unless they are completely documented. Documentation is also required to establish effective maintenance procedures for the solution. This is related to professional ethics.

1.4.4. DATA

A very important activity of the Operations Research methodology is the collection of data, which does not appear in any of the eleven phases described because it does not take place at any specific point of the analysis as an independent step. We begin to collect data and to identify their sources from the moment we make contact with the case to be studied. As we move on, we may need more data to describe the relevant system. For some projects most of the data are available when the mathematical model is built. The specific form of the quantitative relationships can only be determined if we know the main characteristics of the data, such as whether the relationships are linear or nonlinear. In some cases data have to be integrated to mathematical relationships. In other cases, the main part of the data can wait until the model is ready for implementation.

In summary, the identification of the data sources, data collection and evaluation are activities that may happen parallel to any of the eleven steps of the methodology described, even in the last step of the revision of the solution. In some cases data may not be available in the required form or may even not exist. In these cases, actions should be aimed at the collection of data in the required form.

Lastly, we would also like to emphasize the importance of the analyst's skills regarding personal relationships and their ability to obtain information from interviews. Open interviews are recommended to perform, showing curiosity and interest in what others know, rather than giving the image of an expert who knows everything.

1.5. SUMMARY

This chapter approaches the basic nature and methodology of Operations Research in order to understand and assess the role of Operations Research in the education and training of Graduates in Business Administration and Management. Operations Research is a ***technology*** whose purpose is the production of information about systems to improve the efficiency of companies and other organizations. The **construction** and the **solving techniques of mathematical models** are only a part of a real Operations Research project. The **Problem formulation** and **solution implementation** phases are also key, so we should not lose sight of the role played by the model and the solution within the methodology of Operational Research.

The aim of the remaining chapters is to facilitate learning of the formulation and solving linear programming models, integer, nonlinear and multiple criteria using Microsoft Excel *Solver* and LINGO and other professional software (*Expert Choice* and *D-Sight*). We will focus on the key concepts and the necessary techniques for the correct interpretation of the results of the models, in order to improve decision-making and ultimately the effectiveness of companies. We recommend that the student read this introductory chapter again at the end of the course, when they will better understand some of the issues discussed.

1.6. SELECTED REFERENCES

1. Ackoff, R.L. (1987): *The Art of Problem Solving: accompanied by Ackoff's Fables*. John Wiley & Sons.

2. Assad, A.A., Wasil, E.A. and G.L. Lilien (1992): *Excellence in Management Science Practice. A readings book*. Prentice-Hall International.

3. Daellenbach, H.G.; George, J.A. and D.C. McNickle, (1987): Introducción a las técnicas de Investigación de Operaciones. CECSA.

4. Daellenbach, H., McNickle, D. and S. Dye (2012): Management Science. Decision-making through systems thinking. 2nd Revised Edition. Palgrave Macmillan.

5. Eppen, G.D.; Gould, F.J.; Schmidt, C.P.; Moore, J.H. and L.R. Weatherford (1997): Introductory Management Science: Decision Modeling with Spreadsheets. 5th edition. Pearson.

6. Hillier, F.S and Hillier, M.S. (2011): Introduction to Management Science. A Modelling and Case Studies. Approach with Spreadsheets. Fourth Edition. McGraw-Hill.

7. Hillier, F.S. and Lieberman G.J. (2010): Introduction to Operations Research. Ninth Edition. McGraw-Hill.

8. Keys, P. (Ed) (1995): Understanding the process of Operational Research. Wiley.

9. Kirby, M.W. (2000): “Operations Research Trajectories: The Anglo-American Experience from the 1940’s to the 1990’s”, Operations Research 48, 661-670.

10. Robinson, R. (2000): “A Business Executive´s Guide to Modern OR”, OR/MS Today 27/3, 22-27.

11. Weintraub, A.; Romero, C.; Bjorndal, T. and R. Epstein (2007) (Editors): Handbook of Operations Research in Natural Resources. Springer.

12. Winston, W. L. and S.C. Albright (2012): Practical Management Science. Fourth Edition. South-Western. USA.

CHAPTER 2

FORMULATING AND SOLVING LINEAR PROGRAMMING MODELS: BASIC CONCEPTS

Linear Programming is the most important technique used in Operations Research and is considered to be as one of the most significant scientific advances of the 20th century. It is a standard *tool for solving optimization problems* that has had an extraordinary impact since 1950 and currently saves millions of dollars for many companies and businesses. Some of the most common applications include problems such as allocating production resources, material blending, distribution, transportation, food planning and radiation therapy design.

In linear programming real problems are represented by mathematical models subject to a number of conditions, such as the linear nature of the functions. Similar to what happens with other Operations Research techniques, model construction is an essential stage and it is mainly the fruit of experience and the correct application of some basic principles. We begin this chapter by formulating a problem and then a linear programming model in order to solve it step by step. In this way all the assumptions of linear programming models will be explained. We will also see that the meaning of programming in this context is rather different from the term "programming", as used in computer science to refer to software implementation.

Further, we will solve graphically one problem to help the intuitive understanding of the main basic concepts, such as feasible solution, feasible region and optimal solution. The concept of slack variables and the effects of changing the parameters of the model through sensitivity analysis are also presented.

Next, we will solve the problem as we would in practice, using spreadsheets or optimization software and we will interpret the results. The chapter includes other problems that will facilitate learning the modelling and subsequent resolution and interpretation of the solution to improve decision-making in business. Finally, selected references on this topic are given as a guide to provide students with extra reading material. The chapter includes case studies; some of which will be used in laboratory sessions and solved by the students with the help of the teacher, and other case studies will serve as self-assessment exercises.

2.1. THE PROBLEM: PRODUCTION IN A POWER PLANT AND POLLUTION CONTROL

The management of a coal-fueled thermal power plant is analyzing the operational configuration of the plant to adapt to new environmental pollution control regulations. The maximum emission rates of the thermal power plant are

- Maximum emission of sulphur oxide: 3000 parts per million (PPM)
- Maximum emission of particles (smoke): 12 kilograms/hour (kg/h)

The coal is transported to the plant by train and is unloaded into containers close to the plant. A conveyor belt takes it to the pulverizer, in which it is handled and fed directly to the combustion chamber at the correct speed. The heat generated in the combustion chamber is used to produce steam, which, in turn, serves to drive the turbines.

Two types of coal are used: type A, a hard, clean-burning coal with low sulphur content (rather expensive); and type B, a cheap, relatively mild coal with high sulphur content that causes smoke, as shown in Table 2.1. The thermal value in terms of the steam generated is greater for coal type A than for type B, which are 24000 and 20000 lb per ton respectively.

Table 2.1. Emission of polluting agents

Coal	Sulphur oxide in fuel gases	Particles (kg emission/ton)
A	1800 PPM	0.5 Kg/ton
B	3800 PPM	1.0 Kg/ton

As coal type A is hard, the pulverizer can only handle 16 tons of coal A per hour; however, it can handle up to 24 tons of coal B per hour. The loading system of the conveyor belt has a capacity of 20 tons per hour independent of the type of coal.

One of the many questions the plant's management has given is the emission limits of the polluting agents and the types of coal available, what is the maximum possible amount of electricity that can be generate in the power plant? The answer will allow managers to determine the safety range in order to cover peak power demands.

2.2. THE MODEL: VARIABLES, OBJECTIVE FUNCTION AND CONSTRAINTS

2.2.1. VARIABLES: DIVISIBILITY AND NONNEGATIVITY HYPOTHESIS

In the short term, the installations of the power plant are fixed. The only aspect of the problem that can be changed and used to modify the production of the power plant is the amount of each type of coal to burn. Therefore, the decision variables of the problem are

- Amount of coal A used per hour, referred to as X_1 (ton/h)
- Amount of coal B used per hour, referred to as X_2 (ton/h)

Linear programming often refers to the controllable aspects of decision-making problems as activities. Therefore, X_1 and X_2 represent the burning activity levels of coal A and coal B, respectively.

LINEAR PROGRAMMING HYPOTHESIS 1: DIVISIBILITY

All variables can take any real value

Many activities of the real world vary in a continuous way, i.e., they are divisible infinitely. For example, the amount of coal burnt per hour can be adjusted within certain limits. However, there are real activities that can only take integer values, for example, the number of truck trips necessary to move a certain load from one place to another or the number of computers required by a company.

When the real activity is not divisible in a finite way, but the normal level of the activity is a high number, then divisibility conditions can be used as a convenient approach. This means that the value of the solution is ten or higher. Fractional values are rounded up to the closest integer value. However, if the normal level of the activity is less than 10, *integer programming* will be necessary.

LINEAR PROGRAMMING HYPOTHESIS 2: NONNEGATIVITY CONDITIONS

All variables are nonnegative

This hypothesis reflects the nature of most real activities, since negative activity levels hardly ever occur in economic or engineering contexts. However, this consideration does not involve a loss of generalization. Any number (positive, zero or negative) can be expressed as the algebraic difference between two nonnegative numbers. If an activity can occur in both positive and negative levels (for example, buying or selling bonds), two variables are introduced for this activity, X^+ for nonnegative levels, and X^- for nonpositive levels. Their difference $X = X^+ - X^-$ represents the real level of the activity. With this method, both X^+ and X^- are subject to be nonnegative. Optimization software allows users to define directly these variables as free variables with a variation range between negative and positive infinity.

2.2.2. OBJECTIVE FUNCTION AND CONSTRAINTS: LINEARITY HYPOTHESIS

The objective of the plant's management is to maximize power generation in the plant. Since electric power is generated from steam there is a direct relationship between steam and electric power generation, maximizing steam generation is equivalent to maximizing electric power generation. Therefore, the management objective can be restated as "finding the combination of fuels that maximizes steam generation ".

How much steam is produced for any given amount of coal used? A simple and systematic way of determining this is shown in Table 2.2.

Let us express the amount of steam generated in thousands of pounds. Therefore, coal A produces 24 steam units and coal B 20 steam units per ton of coal. Thus, the amount of steam generated per hour is

$$(1)\quad 24\,X_1 + 20\,X_2 = Z$$

Table 2.2. Building the objective function

Coal	*Stream (lb/ton)*	*Coal used (ton/hour)*	*Stream generated (lb/hour)*
A	24000	X_1	24000 X_1
B	20000	X_2	20000 X_2
Total amount of stream lb/h = 24000 X_1 + 20000 X_2			

The first term in (1) is called the objective function and Z is the value of the objective function. The variable coefficients are called objective function coefficients. The problem requires determining values of X_1 and X_2 that maximize Z value. Figure 2.1. shows that (1) is a family of parallel straight lines and that for each value of Z we obtain a straight line, whose points represent the possible combinations of X_1 and X_2 that generate the same amount of steam and thus, of energy. For this reason, they are known as isoproduction lines (isoprofit or isocost, in the case that the objective function corresponds to profit or cost respectively). It can also be noted that the objective function is linear.

LINEAR PROGRAMMING HYPOTHESIS 3: LINEARITY

All relationships between variables are linear. In linear programming this implies:

1. **Proportionality of the contributions. The individual contribution of each variable is strictly proportional to its value, and the proportionality factor is constant for the range of values that the variable may take.**

2. **Additivity of the contributions. The total contribution of the variables is equal to the sum of the individual contribution, regardless of the value of the variables.**

A relationship such as $Z = 5\ X_1 + 3\ X_1^2 + 2\ X_2$ or $Z = 24\ X_1 + 20\ X_2$ for $X_1 \leq 5$ and $10 + 22\ X_1 + 20\ X_2$ for $X_1 > 5$ would violate the condition of proportionality; whereas $Z = 24\ X_1$ for $X_2 = 0$, $20\ X_2$ for $X_1 = 0$ and $22\ X_1 + 18\ X_2$ for $X_1 > 0$ and $X_2 > 0$ would violate the condition of additivity.

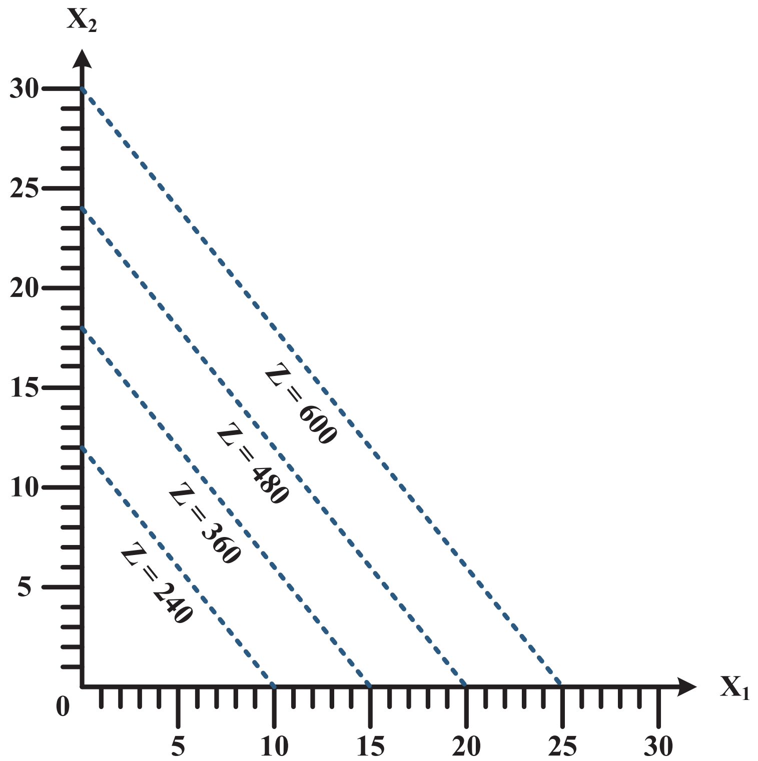


Figure 2.1. Objective function

Hypothesis 3 implies constant scale profits and prevents scale economies. In practice, this condition probably does not strictly hold; in particular, for very low or very high activity values. However, if this condition is fulfilled in an approximate way within the normal range of solution values, a linear programming model would be a good approach. This consideration also excludes the problem of fixed costs when they are presented for positive values of the variable, but not for zero values.

In addition to the nonnegativity conditions, the values of the variables should fulfill certain constraints that may be of a physical, economic or legal nature.

Constraint of particle emissions

The maximum amount of smoke emissions per hour in power plants is limited to 12 kg. According to Table 2.1, one ton of coal A produces 0.5 kg smoke and one ton of coal B produces 1 kg smoke. If the plant burns X_1 tons of coal A and X_2 of B, the total amount of smoke emitted by both types of coal is equal to the sum of both, which cannot exceed 12 kg/h.

$$(2) \quad 0.5\, X_1 + X_2 \leq 12$$

The coefficients of the variables in the constraints are called *technical coefficients* and the second term of the inequality or independent term is known as *right-hand side* (RHS) constraint parameter.

Constraint of loading installations

The system of the conveyor belt that transports the coal from the containers to the pulverizer has a capacity of 20 ton/h. Therefore, the load constraint will be:

$$(3) \quad X1 + X2 \leq 20$$

Constraint of the pulverizer capacity

The maximum capacity of the pulverizer is 16 ton/h for coal A or 24 ton/h for coal B. That is, it takes 1/16 h to handle one ton of coal A and 1/24 h to pulverize one ton of coal B. If the solution demands the combination of both types of coal, the time required to pulverize a mixture of X_1 ton of A and X_2 of B is $(1/16)X_1 + (1/24)\, X_2$. This constraint is expressed as a combination of X_1 and X_2 for 1 hour. Therefore, the constraint of the pulverizer is:

$$\frac{1}{16} X_1 + \frac{1}{24} X_2 \leq 1$$

or

$$(4) \quad 1.5\, X_1 + X_2 \leq 24$$

Note how the difficulty arising from the different maximum rates has been solved. The rates have been converted to required time per ton and the constraint is expressed terms of time rather than capacity.

Constraint of sulphur oxide emissions

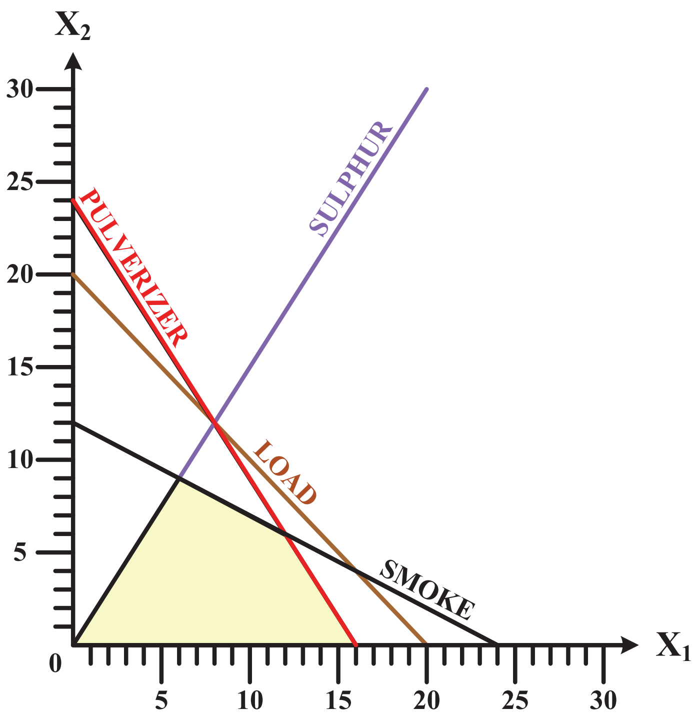
Maximum sulphur oxide emissions should not exceed 3000 PPM at any time. Since both types of coal are burnt simultaneously, the combination of X_1 ton of coal A, and X_2 tons of coal B per hour that feeds the combustion chamber is considered as an homogeneous mixture.

$X_1/(X_1 + X_2)$ of the mixture is coal A with a sulphur oxide emission rate of 1800 PPM and $X_2/(X_1 + X_2)$ of the mixture is coal B, with a sulphur oxide emission rate of 3800 PPM. The emission rate of the mixture will be the weighted mean value of the individual emission rates, using the fractions of each type of coal as weighting values. The weighted mean value should not exceed 3000 PPM:

$$1800\frac{X_1}{X_1+X_2}+3800\frac{X_2}{X_1+X_2}\leq 3000$$

By multiplying both sides of the inequality by ($X_1 + X_2$) and reordering the terms, we obtain the constraint:

$$(5)\quad 1200\,X_1 - 800\,X_2 \geq 0$$



Figue 2.2. Feasible Region

Figure 2.2 illustrates these four constraints. Thus, to draw the smoke constraint, we just have to put a variable to zero, for example X_1 in the equation of line (2) and obtain the value of $X_2 = 12$. In the same way, $X_2 = 0$ in the smoke constraint line and by solving the equation we obtain $X_1 = 12/0.5 = 24$. We represent the line for the smoke constraint connecting the points (0, 12) and (24, 0).

2.2.3. GENERAL FORMULATION OF A LINEAR PROGRAMMING MODEL: CERTAINTY HYPOTHESIS

To sum up the mathematical model that we have formulated to represent the problem stated above is the following:

- To determine the values of the **variables**: $X_1 \geq 0$ and $X_2 \geq 0$
- These optimize, maximize in this case (in other cases, minimize) **the objective function**:

$$Max\ 24\ X_1 + 20\ X_2$$

- and that they meet **the constraints**:

$$0.5\ X_1 + X_2 \leq 12 \qquad (smoke)$$

$$X_1 + X_2 \leq 20 \qquad (load)$$

$$1.5\ X_1 + X_2 \leq 24 \qquad (pulverizer)$$

$$1200\ X_1 - 800\ X_2 \geq 0 \qquad (sulphur)$$

The general formulation of a linear programming model is the following:

Determine the values of the *decision variables* $X_j \geq 0$ *for* $j = 1, 2, ..., n$ which optimize (maximize or minimize) the *objective function:*

$$MAX \sum_{j=1}^{n} c_j X_j$$

Subject to

$$\sum_{j=1}^{n} a_{ij} X_j \ \leq, = o \geq b_i \quad for\ i = 1,2, \dots m$$

Where n is the number of variables and m is the number of constraints.They can be equalities or inequalities of type ≤ or ≥. c_j, a_{ij} and b_i are the parameters of the model.

LINEAR PROGRAMMING HYPOTHESIS 4: CERTAINTY

All parameters of the model c_j, a_{ij} and b_i are known constants.

2.3. FEASIBLE REGION AND GRAPHICAL SOLUTION

For a solution to be possible, the combination of activity levels must simultaneously satisfy all the constraints, including the nonnegativity conditions. This solution is called the ***feasible solution*** for the problem. The set of all feasible solutions forms the ***feasible region*** or set of possible solutions. Observe in Figure 2.2 that the set of possible solutions does not depend on the objective function. This is an interesting property of most Operations Research models and has important consequences on the resolution method and the properties of the optimal solution.

If the boundary of a certain constraint has no points in common with the feasible region, then this **constraint** is ***redundant*** and can be removed from further consideration, since it will never limit the values of the variables. Is there a redundant constraint in our problem?

In practice when a problem presents hundreds of constraints and hundreds of variables, it is difficult to identify whether a constraint is redundant or not. Fortunately, the resolution algorithm known as the simplex method works efficiently, even if the formulation of the problem contains redundant constraints.

Since the main purpose is to maximize the generation of steam in the power plant, we will have to determine the highest line containing at least one possible solution. This line is the line corresponding to $Z = 408$ and the activity levels of the variables in the optimal solution are $X_1 = 12$ and $X_2 = 6$, as shown in Figure 2.3. The combination of 12 tons of coal A and 6 tons of coal B per hour maximizes the production of steam in the power plant within the physical and legal constraints of the variables.

From an intuitive point of view, it seems obvious that the optimal solution will always occur at the limits of the feasible region, either at a corner-point or on one side of the polygon. As explained in sensitivity analysis, the slope of the objective function determines the location of the optimal solution.

If the problem demanded the minimization of the objective function, how would the graphical method change to find the optimal solution? For example, we want to determine the solution of minimum cost to obtain a steam production of at least 216 units per hour with a cost per ton of 24 euros for coal A and 15 for B. Formulate and solve this problem graphically.

To sum up we can say that an ***optimal solution*** is a **feasible solution** with the **best value** of the **objective function**. The best value of the most suitable values of the objective function is the highest value in maximization problems or the lowest value in minimization problems.

Not all linear programming problems present a happy ending. On the one hand, it may happen that the constraints are inconsistent in the sense that there is no feasible solution. On the other hand, the feasible region can be open in one direction so that the objective function may increase indefinitely without a finite solution (unbounded solution). These cases are not frequent in practice. These solutions often appear as a result of errors or incorrect representations of the problem in the mathematical formulation. Therefore, when solving a linear programming model we may find four cases:

1. **Unique solution**.
2. **Alternative solutions** (infinite solutions). In a model with two variables it always happens that the objective function cuts the feasible set on one side of the polygon, for the best value. In practice we will see in following chapters that this is a rather common case in real problems.
3. **There is no solution**, because none of the combinations of the variables fulfills all the constraints.
4. **Unbounded solution**.

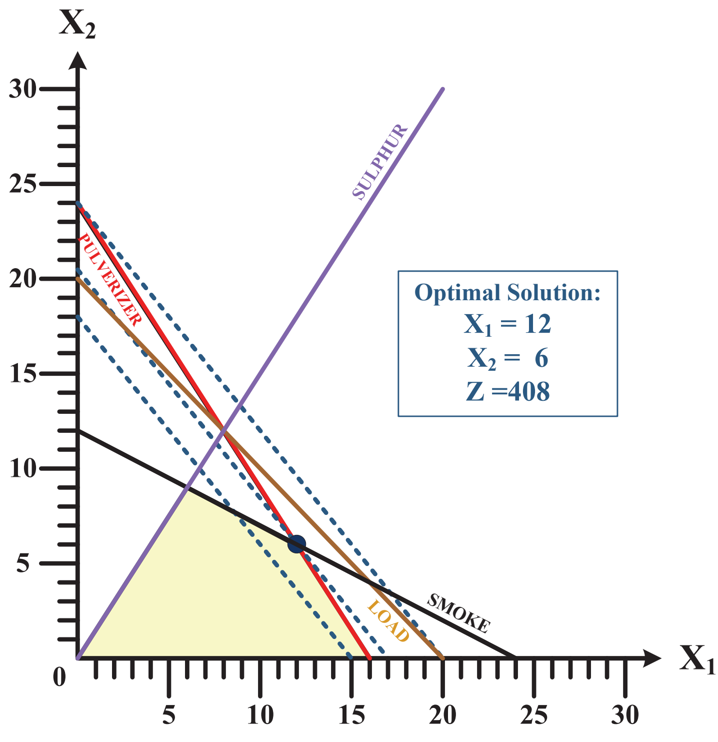


Figura 2.3. Optimal solution

2.4. SLACK VARIABLES

For any feasible solution, the difference between the value of the constraint and the coefficient of the second member is called slackness (for inequalities≤) or surplus (for inequalities ≥). It is often convenient to show this difference explicitly, introducing an additional variable to each constraint. These variables are called slack or surplus variables. In short, the term slack variable is used to express both variables. These variables are subject to the same conditions of divisibility and nonnegativity as the decision variables. Thus, each constraint becomes equality. Introducing the slack variables in our example:

$$0.5\,X_1 + X_2 + X_3 = 12\,(\text{smooke})$$

$$X_1 + X_2 + X_4 = 20\,(\text{load})$$

$$1.5\,X_1 + X_2 + X_5 = 24\,(\text{pulverizer})$$

$$1200\,X_1 - 800X_2 - X_6 = 0\,(\text{sulphur})$$

Why is the slack variable subtracted in the sulphur constraint, instead of added as in the other constraints?

Slack variables can often be interpreted as non-used resources or non-used capacity for a given solution. For example, X_3 is the amount of non-used smoke emission capacity, and X_4 the amount of non-used load capacity. Which is the interpretation corresponding to X_5? Due to the way in which the sulphur constraint was obtained there is no simple interpretation for X_6. Check that X_3 and X_5 are zero in our example, whereas $X_4 = 2$ and $X_6 = 9600$.

2.5. SENSITIVITY ANALYSIS

2.5.1. SENSITIVITY ANALYSIS OF THE OBJECTIVE FUNCTION COEFFICIENTS

We mentioned earlier that the slope of the objective function determines the place where the optimal solution is located. The slope of the objective function is given by its coefficients. What happens if one of these coefficients changes? Suppose that the thermal value of coal A is 36000 pounds of steam, instead of 24000, all other coefficients remaining constant. Then, the objective function will be

$$MAX\ 36\,X_1 + 20\,X_2$$

This new objective function is shown in Figure 2.4, together with the new optimal solution, which consists of using only one type of coal, specifically 16 ton/h of coal A that produces 576 thousand pounds of steam. Therefore, a change in the value of one coefficient of the objective function (with all other coefficients remaining constant) causes changes in the objective function slope, and if this change is large enough, the optimal solution moves to another corner-point of the feasible set. In this particular case, it has shifted from point A to point B.

Let us see this effect in more detail. What is the highest possible value of the original coefficient of X_1 in the objective function C_1, prior to the change in the optimal solution from point A to point B? It can be observed that as C_1 increases, the objective function slope comes closer to the slope of the pulverizer constraint line until they finally coincide. For this value of C_1, any point along the line going from A to B is optimal; that is, alternative optimal solutions are obtained, all of them with the same Z value. In many real problems alternative optimal solutions occur naturally. If C_1 increases a little more, then the optimal solution will only be point B.

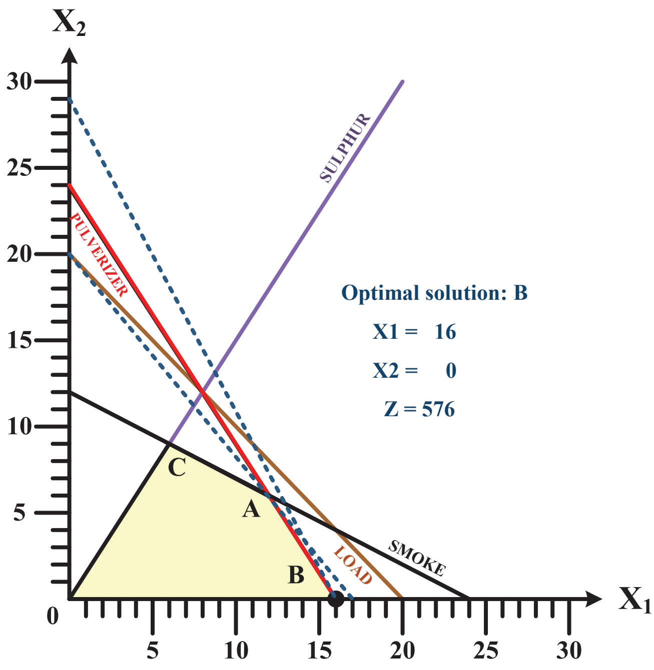


Figure 2.4. Alternative objective function

To calculate the range of values between which C_1can be modified, A being the optimal solution (Figure 2.4), we only have to match the slopes of the objective function and constraints that intersect point A. The objective function and the constraint of the pulverizer are parallel when their slopes are equal. This implies that the ratio between the coefficients of X_1 on the line of the objective function and the constraint of the pulverizer is equal to the ratio of the coefficients of X_2 in both equations.

Objective Function: $\boldsymbol{C_1} X_1 + \mathbf{20}\, X_2$

Pulverizer: $\mathbf{1.5}\, X_1 + X_2 = 24$

$$\frac{C_1}{1.5} = \frac{20}{1} = 20$$

This gives $C_1 = 30$. For an increase in C_1 higher than 30, the optimal solution will change from A to B. For $C_1 = 30$ the objective function $30\,X_1 + 20\,X_2$ takes its maximum value of 480 at every point of line AB. Similarly, if C_1 decreases to 10 (all other coefficients remaining constant) the objective function will be parallel to the smoke constraint. If C_1 were lower than 10, the optimal solution would change from point A to point C (see Figure 2.4).

In conclusion, we can say that if the coefficient C_1 is such, that the slope of the line that represents it is between the slope of the constraints of the pulverizer and smoke, the initial optimal solution (point A) is optimum. And we have deduced that if the rest are constant, the solution $X_1 = 12$ and $X_2 = 6$ will be the optimal solution for any value of the X_1 coefficient of the objective function for the interval $10 \leq C_1 \leq 30$. Can you determine the interval corresponding to C_2 without changes in the initial optimal solution, i.e. point A in Figure 2.4?

2.5.2. SENSITIVITY ANALYSIS OF THE RIGHT-HAND SIDE OF THE CONSTRAINTS

Let us see what happens to the optimal solution when the right-hand side of a constraint changes. Suppose that the management is considering the installation of a system that reduces the amount of smoke emissions by 25%. This will allow the plant to fulfill the legal regulations while emitting up to 15 Kg/h of uncontrolled smoke from the combustion chamber. What would be the effect in terms of increasing steam generation?

Let us first consider that the maximum allowed smoke emissions increase from 12 to 13 kg/h, all the other coefficients remaining constant. This causes a move upwards in the smoke constraint. Figure 2.5 shows how the feasible region increases. In the new feasible region Z = 408 is not the optimal value of the objective function, as its best value lies at point D. Therefore, the optimal solution changes from A to D. This change occurs due to the fact that the smoke constraint is strictly fulfilled in the optimal solution of the original problem. Now, the new activity levels of the variables are $X_1 = 11$ and $X_2 = 7.5$. The decrease in X_1 causes a reduction of 24 steam units, whereas the increase in X_2 increases steam production by 30 units. The net increase is 6. Thus the new maximum value of the objective function will be Z = 408 + 6 = 414.

The improvement of the optimal value of the objective function due to the unit increase in the RHS of a constraint is called ***opportunity cost or dual price*** of the constraint. In this case the opportunity cost of the smoke constraint is 6.

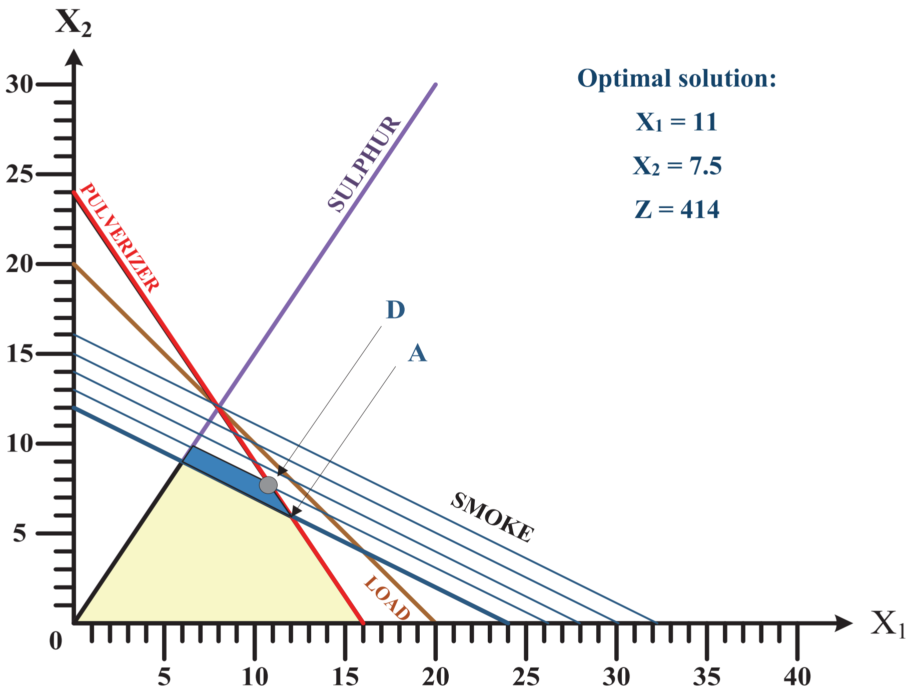


Figure 2.5. Sensitivity analysis of the right-hand side of smoke constraint

What would happen if the maximum smoke emissions were 14, 15, 16 and 17 kg/h? Figure 2.5 shows how the area of the feasible region increases with each change to a maximum of 16. Check that for each change the objective function increases by 6.

For an increase higher than 16, the smoke constraint becomes redundant. Now the optimal solution will be restricted by the pulverizer, sulphur and load constraints. Thus, the opportunity cost of this constraint is zero for values higher than 16.

The original question required the determination of the increase in the production of steam due to the change in the allowable smoke levels from 12 to 15 kg/h. This will be 3 x 6 = 18 steam units/h.

What is the opportunity cost of a constraint which is not strictly met in the optimal solution? It becomes clear that if one part of the resource is not used, i.e. the slack variable is positive, the additional amounts of that resource have no value. They would only increase the amount of slack. Therefore, the dual price of that constraint is zero. Determine the dual prices of the other constraints. Observe the **relationship** between the

opportunity cost of a constraint and the **slack variable** associated with it. When one resource is completely used its opportunity cost is generally positive (nonnegative to be more exact) and its slack variable is zero, whereas when the slack variable is positive the dual price is zero.

Dual prices provide management with valuable information about the profits that can be obtained by smoothing the constraints. If the benefits exceed the cost generated by smoothing a given constraint, then the changes are attractive.

2.6. THE EXTENDED PROBLEM: A NEW VARIABLE

The power plant is offered a third type of fuel, coal type C, that has a sulphur oxide emission rate of 2000 PPM, a smoke emission rate of 0.8 kg/ton of burnt fuel, and requires 1/20 h per ton of the pulverizer and loading capacity. Its thermal value is equivalent to 21000 lb of steam per fuel ton. Is it profitable to use this fuel in the plant?

Let us reformulate the problem with this third type of coal. Let X_3 be the number of coal C tons per hour. Thus

$$Max\ 24\ X_1 + 20\ X_2 + 21\ X_3$$

Subject to

$$\begin{array}{ll} 0.5\ X_1 + X_2 + 0.8\ X_3 \leq 12 & (smoke) \\ X_1 + X_2 + X_3 \leq 20 & (load) \\ 1.5\ X_1 + X_2 + 1.2\ X_3 \leq 24 & (pulverizer) \\ 1200\ X_1 - 800\ X_2 + 1000\ X_3 \geq 0 & (sulphur) \\ X_1 \geq 0,\ X_2 \geq 0 \text{ and } X_3 \geq 0 & \end{array}$$

The dual prices of the original problem provide all the information needed to know if we are interested in this new coal. If we decide to use one ton of coal C, $X_3 = 1$, we must have the required machine capacity (loading systems and pulverizer) and have the possibility to emit smoke and sulphur that would be generated with it. This is equivalent to reducing the right-hand side of the constraints of the original problem in the following way:

$$\begin{array}{ll} 0.5\ X_1 + X_2 \leq (12-0.8) \text{ or } 11.2 & (smoke) \\ X_1 + X_2 \leq (20-1) \text{ or } 19 & (load) \\ 1.5\ X_1 + X_2 \leq (24-1.2) \text{ or } 22.8 & (pulverizer) \\ 1200\ X_1 - 800\ X_2 \geq (0-1000) \text{ or } -1000 & (sulphur) \end{array}$$

The loading system allows us to use a ton of the new type of coal, as we have spare capacity. We can also emit more sulphur, since the latter constraint also has slack. Therefore, the opportunity costs of these two constraints are zero.

However, as we can not emit more smoke and we do not have more capacity in the pulverizer, we can only burn a ton of new coal C if we do not burn any coal A and/or B. In this way we have the pulverizer available and the possibility of emitting the smoke needed to burn a ton of coal C.

Specifically, to burn one ton of coal C we need to reduce the use of the pulverizer by 1.2 (decreases the RHS). As the opportunity cost is 14, the result will cause a decrease of 1.2 x 14 = 16.8 in the value of the objective function. Similarly, to burn a ton of coal C, we have to stop emitting 0.8 kg of smoke from burning coal A and B. A reduction of 0.8 kg of maximun emission of smoke decreases the value of the objective function by 0.8 x 6 = 4.8. The total decrease of the objective function value is equal to their sum, 21.6 steam units. Futhermore, the additional output per hour obtained by burning one ton of coal C is only 21 units. Thus, the net loss in steam production is 0.6 units. Therefore, in these conditions it is not advantageous to use coal C and the optimal solution remains the same.

2.7. LINEAR PROGRAMMING MODEL SOLVING WITH A SPREADSHEET

The graphical solution is only possible if the number of variables is not higher than 2 (3?). Problems with more variables will have to be solved mathematically; for example, by applying the simplex method (an efficient algorithm that will be explained in chapter 3). As real problems have hundreds or thousands of variables and constraints, in practice the problem is solved using optimization software. Since spreadsheets are the most used tools in the business environment we will see its performance for solving linear, integer and nonlinear programming models. **Annex 1** explains in detail the procedure for entering data for a linear programming model and solving it in Excel. Table 2.3 presents the data of the problem and the optimal solution. The value of the decision variables, the objective function and the first member of the constraints in the optimal solution are presented in italics. The remaining data are the model coefficients.

Tables 2.4 and 2.5 are the reports generated by the Excel Solver tool after solving the model and pick the optimal solution and sensitivity analysis respectively. In Table 2.4 you can see the values of the variables and the objective function at the optimal solution, and the value of the slack variables for the constraints. The column "Status" shows "Binding" to indicate that in this case the constraint is checked strictly, ie the slack variable is zero and "Not Binding" when the slack variable is positive.

Table 2.3. Model and Optimal Solution of the problem of energy production and pollution control

	A	B	C	D	E	F	G	H	I
1	ENERGY PRODUCTION AND POLLUTION CONTROL								
2									
3					**Coal A**	**Coal B**			
4	Steam production in thousands of lb/ton				24	20			
5							**Used capacity LHS**		**RHS**
6	Emission of smoke kg/h				0.5	1	12	≤	12
7	Loading installation				1	1	18	≤	20
8	Pulverizer capacity				1.5	1	24	≤	24
9	Emission of sulphur				1200	-800	9600	≥	0
10									
11					**Coal A ton/h**	**Coal B ton/h**			**Total Steam Production thousand lb/h**
12					12	6			408

Table 2.4. Optimal solution to the problem of energy production and pollution control

Microsoft Excel 14.0 Answer Report

Objective Cell (Max)

Cell	Name	Original Value	Final Value
$I$12	TotalSteamProduction	0	408

Variable Cells

Cell	Name	Original Value	Final Value	Integer
$E$12	Coal A (ton/h)	0	12	Contin
$F$12	Coal B (ton/h)	0	6	Contin

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$G$6	Emission of smoke kg/h	12	$G$6<=$I$6	Binding	0
$G$7	Loading installation	18	$G$7<=$I$7	Not Binding	2
$G$8	Pulverizer capacity	24	$G$8<=$I$8	Binding	0
$G$9	Emission of sulphur oxide	9600	$G$9>=$I$9	Not Binding	9600

Table 2.5. Sensitivity analysis for the problem of energy production and pollution control

Microsoft Excel 14.0 Sensitivity Report

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$E$12	Coal A (ton/h)	12	0	24	6	14
$F$12	Coal B (ton/h)	6	0	20	28	4

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$G$6	Emission of smoke kg/h	12	6	12	4	4
$G$7	Loading installation	18	0	20	1E+30	2
$G$8	Pulverizer capacity	24	14	24	4	6
$G$9	Emission of sulphur oxide	9600	0	0	9600	1E+30

Table 2.5 is the sensitivity analysis. Firstly it indicates the range over which each coefficient in the objective function can vary without changes to the optimal solution. For example, the coefficient of X_1, which can increase by 6 and decrease by 14, ie, can be between 10 and 30. It also gives us the ***reduced cost*** of variables, which is an important concept and can be interpreted as the amount that should improve the objective function coefficient of the variable sufficiently for it to take a nonzero value in the optimal solution. In this case, as the variables have a positive value, its reduced cost is zero. When the variable has a positive value, because it has a lower bound greater than zero, the interpretation of the reduced cost is penalty, in terms of the objective function to introduce the variable in the solution.

Secondly the sensitivity analysis provides the range of variation of the right-hand side of the constraint that does not change the value of the opportunity cost. For example, the RHS of the restriction of smoke worth 12 kg/hour can be between 8 and 16, as it can increase by 4 and decrease by 4, without changes in the value of 6 for its opportunity cost. 6 is the increase in steam production (objective function) for additional unit on the RHS, i.e. per each kg/hour more of smoke that can be generated. Please read Annex 1 for detailed information.

In summary, to solve a linear programming model with Excel, we introduce the problem **data** in data cells, which correspond to the **technical coefficients (a_{ij})**, **the objective function coefficients (C_j)** and **the right-hand side of the constraints (b_i or RHS)**. After **variable cells** are defined, where we have the values of **decision variables, the linear functions** are introduced that represent the **constraints** and the **objective function**. The SUMPRODUCT function of the spreadsheet is useful to introduce linear functions of the model (objective function and constraints). The use of range names in formulas is also interesting to simplify the data entry process and improve the understanding of the model (see Annex 1).

The features of Excel for solving optimization models have improved greatly in recent years. In fact, they have incorporated latest and advanced methods such as genetic algorithms that we explain in the last chapter of the book. However, its use is currently recommended for solving small or medium sized models. For models with thousands of variables and constraints we consider it more appropriate to use more powerful optimization software, including essential modelling languages to generate large models. LINGO and CPLEX are good choices, they also have the ability to import and export data from spreadsheets and databases.

2.8. SOLVING LINEAR PROGRAMMING MODEL WITH OPTIMIZATION SOFTWARE

In this section we highlight the LINDO Systems company that has sold a marketing optimization software for more than two decades, such as ***LINGO*** which incorporates a **model generation language** and **optimizers** to solve **linear, integer, nonlinear and stochastic** programming models. Furthermore, LINDO Systems has a program called *What's Best* which is an add-in to Excel and can solve linear, integer, nonlinear and stochastic programming models with spreadsheets, useful for companies that prefer to use this software environment. Students can download the latest version of LINGO and solving examples and case studies of the book (www.lindo.com). In this website manuals and training material such as the book of Linus Schrage are also available.

The data input of the energy production and pollution control problem, the solution and sensitivity analysis obtained using LINGO are the following:

MODEL:

!EXAMPLE 1: ENERGY PRODUCTION AND POLLUTION CONTROL;

[OBJ] **MAX = 24 * X1 + 20 * X2**;

[SMOKE] **0.5 * X1 + X2 <= 12**;

[LOAD] **X1 + X2 <= 20**;

[PULVERIZER] **1.5 * X1 + X2 <= 24**;

[SULPHUR] **1200 * X1 - 800 * X2 >= 0**;

END

```
Global optimal solution found at step:        5
Objective value:                         408.0000

        Variable           Value        Reduced Cost
              X1        12.00000           0.0000000
              X2         6.00000           0.0000000
```

Row	Slack or Surplus	Dual Price
OBJ	408.0000	1.000000
SMOKE	0.0000	6.000000
LOAD	2.0000	0.000000
PULVERIZER	0.0000	14.000000
SULPHUR	9600.0000	0.000000

Ranges in which the basis is unchanged:

Objective Coefficient Ranges

Variable	Current Coefficient	Allowable Increase	Allowable Decrease
X1	24.00000	6.000000	14.00000
X2	20.00000	28.000000	4.00000

Righthand Side Ranges

Row	Current RHS	Allowable Increase	Allowable Decrease
SMOKE	12.00000	4.000000	4.000000
LOAD	20.00000	INFINITY	2.000000
PULVERIZER	24.00000	4.000000	6.000000
SULPHUR	0.00000	9600.000000	INFINITY

As shown in the input data, we should use the symbol * for multiplication and the semi colon (;) to indicate the end of a sentence, which can be a comment, the objective function or a constraint. Comments start with exclamation marks (!) and the names of the objective function and constraints can be indicated in brackets.

As in Excel Solver we can get a series of reports after solving the model. First comes an overview of the number and type of variables, constraints and coefficients.Then the model provides the optimal value of the objective function, which is 408 in this example. For each variable it indicates the activity level in the optimal solution (X_1 =12 and X_2= 6) and the reduced cost, which is zero in this case. The ***reduced cost*** can be interpreted as the amount by which the coefficient of the objective function of that variable should improve for it to take a value other than zero in the optimal solution. Another possible interpretation is to consider the *reduced cost* as the penalty cost of introducing the variable in the solution.

Next, we obtain the value of the **slack variables *(slack or surplus)*** of the constraints and their opportunity cost or **dual price**. In this section we can observe a result mentioned earlier, that is, the relationship between these two concepts. When the slack variable of a constraint is zero, i.e., the constraint is strictly met, normally its opportunity cost will be other than zero. And when the slack is positive, the associated opportunity cost is zero. Note that this is so in all the constraints and that the first row is not a constraint, but the objective function of the model. Remember that the **opportunity cost** is the amount by which the **objective function improves per unit increase in the RHS of the constraint.**

Thus, for a **maximization** problem when we increase the RHS of a constraint, the new value of the objective function is given by

New optimal value of Z = Former optimal value + Δ RHS* (opportunity cost of constraint)

In the case of **minimization** problems, the term "improve" logically involves decreasing, thus the new optimal value will be

New optimal value of Z = Former optimal value - Δ RHS* (opportunity cost of constraint)

The last two sections correspond to the sensitivity analysis of the objective function coefficients and RHS of the constraints. The first line indicates that the ranges provided are those in which the *basis* does not change. This concept is explained in chapter 3.

Looking at the first part of the table we can say that the coefficient of X_1 in the objective function, which is worth 24, may increase by 6 and decrease by 14 units without changes in the optimal solution. This is without changes in the value of variables X_1 and X_2. Obviously if these values are the same and C_1, which is worth 24, increases or decreases, the value of the objective function that is $C_1*X_1 + 20*X_2$ will change accordingly.

The lower part of the sensitivity analysis refers to the variation of the RHS of the constraints. In the first column we have the constraint name and in the second the value in the model (RHS). The third column indicates the increase of RHS value and the fourth column the decrease, **without changing the value of the opportunity cost** of the constraint. For example, the RHS of the pulverizer constraint is 24 and can increase by four and decrease by six. In other words, its value can be between 18 and 28 without changing its opportunity cost which is 14 (*dual price*). Interpret the ranges provided for other constraints.

2.9. MODELLING: SOME EXAMPLES

2.9.1. COMMON MISTAKES IN MODELLING

There are two extreme ways of learning to build optimization models, one through knowledge of standard examples and other through formulating models creatively. The first option requires much less analytical capability than the second, but it is more limited. It only serves to solve real problems that fit standard models. Obviously, in practice the best approach is to integrate both. The mistakes made in the modelling process can be classified into three categories:

1. Errata or typographical errors
2. Making basic formulation mistakes
3. Approximation errors

The first two types of errors are easy to solve once identified. Typographical errors are harder to find as the model size increases. However, in these cases matrix generators or modelling languages are commonly used, reducing their incidence. Error type 2 is far more serious because it involves not having understood the problem or the formulation of linear programming models.

Errors of type 3 have a more subtle character. In general in developing a model that represents a real situation we need to do some form of approximation. For example, certain products are aggregated, the weekdays are grouped or costs that are not proportional to the variable values are considered linear. To avoid errors of this type one should be able to identify those approximations that are acceptable.

Optimization software usually has capabilities for providing some data about the model, such as the values ranges of the parameters among others. This information is useful in the identification of errors type 1. These errors also tend to provide solutions that are obviously wrong.

Formulation errors are much more difficult to systematize because there are many types. Among the most common is what is known as dimensional analysis: ***the units of all terms of a restriction must be equal***. To avoid this error it is often useful to formulate the problem in words and then write the associated algebraic form.

Another associated error with the measurement units is the use of units in such a way that very large or very small numbers can appear in the same model. This can cause significant rounding errors. This can be avoided by scaling the model in order to reduce the difference between the largest and the smallest coefficient value, and make it as small as possible. Good professional optimization software can solve this problem automatically.

Another formulation error is called non-simultaneity error. In linear programming all constraints must be satisfied simultaneously. We may want to indicate that, if a product is made, it is made to a minimum level, e.g. 20. And the solution indicates if it is made or not, and if yes, the solution will give us its manufacture level. As we shall see, to indicate these situations we should not write $X \geq 20$ and $X \leq 0$. We need to use an integer programming model. *LINGO* can generate the required integer variables and constraints if you indicate that this variable is semicontinuous (See Annex 2).

Finally, we point out that the fundamental characteristics of a **good model** to be useful in decision-making are the following: **simple**, **complete**, **easy to handle**, **adaptable**, **appropriate** to the situation and producing **relevant information for decision-making**. It is advisable to re-read the section on Operational Research methodology explained in Chapter 1.The next section presents a well-known problem to learn how to formulate and solve models, as well as discussing the results obtained. This learning process will continue throughout the book with examples and case studies, many of which students perform with the help of the teacher in laboratory sessions.

2.9.2. SOME MODELS OF LINEAR PROGRAMMING

2.9.2.1. TRANSPORT PROBLEM

It is a frequent problem which arises in the distribution of many products. This problem deals with distributing products from a variety of sources, where there is available supply, to destinations in order to satisfy demands and minimize the cost of transport. In short, the problem is to determine the amount of product to be sent to a number of locations (n) whose demand is $b_1, b_2 \dots b_n$, from a point of origin (m) where the supply is $a_1, a_2 \dots a_m$. The unit cost of shipping from the point of origin i to destination j is known (Cij) and the objective is to minimize the total transport cost.

Take a simple example. It is distributing a product from three factories, with the availability of $a_1=30$, $a_2=25$ and $a_3=21$, to four warehouses, which have requirements of $b_1= 15$, $b_2= 17$, $b_3= 22$ and $b_4=12$. The unit costs between each factory and warehouse are listed in Table 2.6.

Table 2.6. Unit costs of transport from factories to warehouses

Factories/Warehouses	**F1**	**F2**	**F3**
W1	6	4	8
W2	2	9	8
W3	6	5	1
W4	7	3	5

The resulting model for this problem is as follows:

Variables:

X_{ij}: *amount of product to be transported from the factory i to the warehouse j*

Objective function:

$$Min\ Z = 6X_{11} + 2X_{12} + 6X_{13} + 7X_{14} + 4X_{21} + 9X_{22} + 5X_{23} + 3X_{24} + 8X_{31} + 8X_{32} + X_{33} + 5X_{34}$$

Contraints:

Supply F1: $X_{11} + X_{12} + X_{13} + X_{14} \leq 30$

Supply F2: $X_{21} + X_{22} + X_{23} + X_{24} \leq 25$

Supply F3: $X_{31} + X_{32} + X_{33} + X_{34} \leq 21$

Demand W1: $X_{11} + X_{21} + X_{31} \geq 15$

Demand W2: $X_{12} + X_{22} + X_{32} \geq 17$

Demand W3: $X_{13} + X_{23} + X_{33} \geq 22$

Demand W4: $X_{14} + X_{24} + X_{34} \geq 12$

The model has 12 variables (m=3 x n=4) and 7 contraints, 3 supply constraints and 4 demand constraints. The matrix of technical coefficients of the model is shown in Table 2.7. This structure of the technical coefficient matrix of ones and zeros, located in this particular way is that what characterizes the problem of transport and not the context of being a distribution problem. This model can be solved using the general linear programming algorithms that we will see in the next chapter and also another more efficient one for this specific type of problem. The structure of a transport problem ensures a solution with integer values if the supply and the demand of products are integer values too.

Table 2.7. Technical coefficient matrix in a transportation problem

X_{11}	X_{12}	X_{13}	X_{14}	X_{21}	X_{22}	X_{23}	X_{24}	X_{31}	X_{32}	X_{33}	X_{34}	b_i	
1	1	1	1									30	Supply constraints
				1	1	1	1					25	
								1	1	1	1	21	
1				1				1				15	Demand constraints
	1				1				1			17	
		1				1				1		22	
			1				1				1	12	

The general formulation of a transport model is as follows.

Variables

$X_{ij} \geq 0$ amount of product to be transported from the factory i to the warehouse j

Objective function

$$Minimize \sum_{j=1}^{n} \sum_{i=1}^{m} C_{ij}\, X_{ij} \qquad i = 1,2, \dots m \qquad j = 1,2, \dots n$$

The transport problem objective is to minimize the distribution cost.

Constraints

Supply: From each source we cannot send more than the available products

$$\sum_{j=1}^{n} X_{ij} \leq a_i \qquad i = 1,2, \dots m$$

Demand: The demand of each destination b_j should be met

$$\sum_{i=1}^{m} X_{ij} \geq b_j \qquad j = 1,2, \dots n$$

2.9.2.2. OTHER MODELS: LOAD SYSTEMS PLANNING

In a port three types of loading systems are being used to handle four types of cargo: perishable goods, chemicals, minerals and manufactured goods. During the following week the cargo volume that is required to be moved is as follows:

Perishable: 1800 ton

Chemicals: 1500 ton

Minerals: 2000 ton

Manufactured: 1300 ton

Table 2.8 shows the number of tons that can be handled by each loading system per hour, availability of hours of operation system and costs. Loading system 2 cannot mobilize chemical products. Formulate a model for the movement of cargo schedule with the minimum cost.

Table 2.8. Technical and economic data loading systems

LOADING SYSTEM	OPERATING EFFICIENCY (ton/h)				AVAILABILITY IN HOURS	COST €/h
	Perishable goods	**Chemicals**	**Minerals**	**Manufactured goods**		
1	50	100	50	20	500	30
2	20	-	20	20	500	18
3	20	50	50	20	1000	36

2.10. SUMMARY

Linear programming is a tool that allows solving optimization real problems that we represent by a mathematical **model**. First, define the **variables** of the model, which will indicate what the decision makers can control. Then, establish the **objectives** of the problem as a linear function of the decision variables, a function for which we want to find the **optimal value** -maximum or minimum- and the values of the variables which are subject to a number of **constraints**, also represented as linear functions.

Any combination of the model variable values that fulfills all constraints is a ***feasible solution***. The set formed by all feasible solutions constitutes the ***feasible region***, and any feasible point with the best value of the objective function is an **optimal solution**. When solving a linear programming model we may encounter four different cases: a **unique** solution, **alternative** solutions, an **unbounded** solution and **no solution**. The last two cases normally correspond to incorrect formulations of the problem. The first two cases show that the value of the objective function in the optimal solution is unique, but the set of values of the variables giving that best value are not necessarily unique. In real problems, there are many examples that present alternative optimal solutions.

In addition to explaining the basics of formulating and solving linear programming models by the graphic resolution of a simple example, we solved the model using **Excel *Solver*** and ***LINGO*** optimization software. The former is suitable for solving many small and medium sized models and the latter is more recommended for large problems. We have also seen frequent mistakes in formulation and how to avoid them and other important examples of linear programming, such as the transport problem. The case studies are the basis of practical and laboratory classes and a source of self-assessment exercises.

2.11. SELECTED REFERENCES

1. Daellenbach, H.G.; George, J.A. and McNickle, D.C. (1987): *Introducción a las técnicas de Investigación de Operaciones.* CECSA. This excellent book is the source of the example of energy production and pollution control, as well as the case study 3 which have been included in this chapter.

2. Hillier, F.S. and Lieberman G.J. (2010): Introduction to Operations Research. Ninth Edition. McGraw-Hill.

3. Hillier, F.S and Hillier, M.S. (2011): Introduction to Management Science. A Modelling and Case Studies Approach with Spreadsheets. Fourth Edition. McGraw-Hill.

4. LINDO Systems (2011): *LINGO. The modeling language and optimizer*.

5. Winston, W. L. and S.C. Albright (2012): Practical Management Science. Fourth Edition. South-Western. USA.

2.12. CASE STUDIES

CASE STUDY 1: A DIET PROBLEM

Among the problems with more practical applications and in which linear programming has been proven to be very useful are blending problems. Broadly, we can say that a blending problem consists of determining the composition of a product from a number of raw materials in order to maximize profit or to minimize the company's costs. The product must fulfill a number of conditions which are the constraints of the model, together with other constraints that for example may refer to restrictions in the raw materials. Among the industries that actually use this kind of models we can mention metals, fertilizer, food and animal feed industries.

Diet problems are a particular case of blending problems. We can also distinguish between human and animal feeding problems. At present feeding costs in meat production farms for human consumption supposes a high percentage of the overall cost, so that competitiveness in this field depends greatly on the correct calculation of the optimal diet.

For animal breeding, there are two different problems: the calculation of rations and the formulation of animal feed. The former consists of determining the composition of the minimum cost portion that satisfies the animal's nutritional needs. The latter consists of calculating the combination of foods in such a way as to meet a maximum and/or minimum need for nutritional principles per weight unit in order to minimize costs.

The use of high technology is common in many animal farms, among which we find software for calculating optimal feeding. Many of the companies that operate in this sector are multinationals that use linear programming software as a basic tool.

Let us examine a simple example for calculating animal feeding diets that in addition to serving as an illustration of standard linear programming model, can also be used to apply sensitivity analysis. In this type of model it is necessary to perform postoptimal analysis, as the prices of the raw materials usually vary throughout the year. It is also interesting to evaluate the effects of changes in the required protein or energy quantities on the feeding total cost.

Table 2.9 presents the characteristics of six types of raw materials that a farmer needs to obtain the feeding stuff to feed his calves. Their food needs vary with age. Thus the younger calves have different needs than the older calves in the stage prior to slaughtering. Therefore, in fact we have two problems: calculation of the diet for rearing and fattening calves, depending on the age of animals (C1 and C2).

The fibre units (FU) and the digestible protein (DP) must be present in the feeding ration in the minimum quantity indicated. Dry material (DM) can have any value from the range shown in the table. With respect to the amount of concentrated raw materials in the feed, there is the following restriction: for the fattening calves (C2) the FU from concentrated food (barley, soy and sunflower) must be higher than that from non concentrated food (alfalfa, barley and straw); for the baby calves the difference between the FU supplied by the concentrated and the non concentrated food should not exceed 10. Sunflower is limited to be no higher than 10 for rearing calves and to 0.5 for fattened calves due to darkening of the meat.

The problem consists in determining the minimum cost rations for the two types of calves that fulfil the requirements mentioned. The company has formulated two similar blending problems because the matrix of technical coefficients is the same for both. Only the RHS of the constraints differ, the upper and lower value of one of them and the upper bound of one variable.

Using the *Solver* tool of Excel:

1. Determine the minimum cost ration for rearing and fattening calves.
2. Analyze the sensitivity of the diet for the fattened calves to the prices of raw materials and the practical implications of the results.
3. What would be the composition and price of the fattened calves ration if we want it to provide at least 0.65 units of protein? And in the case that the minimum fibre units were 7.6? Answer on the basis of the sensitivity analysis results.

Table 2.9. Technical characteristics of raw materials and diet requirements

Characteristics	Raw materials						Needs	
	Alfalfa	Barley	Wild Barley	Soy	Sunflower	Straw	C1	C2
Fibre units/kg	0.15	1	0.18	1.05	0.88	0.3	>=6.3	>=8.6
Digestible Protein/Kg	0.02	0.06	0.04	0.4	0.28	0.01	>=0.66	>=0.75
Dry Material/Kg	0.22	0.85	0.25	0.9	0.93	0.9	8.8-11.6	8-13
COST (euros/Kg)	0.14	0.30	0.08	0.60	0.42	0.12		

CASE STUDY 2: A FEED PROBLEM

A multinational company in the food sector has several feed factories in the country. The feed formulations which the company manufactures and distributes is calculated every month in the central headquarters taking into account the prices and availability of raw materials. Table 2.10 shows the data to solve a small example of this real problem.

The company needs to calculate the optimal formulation in order to minimize the cost of feed for a certain type of animal, whose nutritional requirements are the following: The feed must have a protein percentage between a minimum of 12% and a maximum of 15%, while the minimum levels of calcium and phosphorus should be 1 and 0.30% respectively.

Table 2.10. Characteristics of raw materials and feed requirements

Characteristics	Corn	Wheat	Barley	Alfalfa
Price (euros/ton)	142	134	125	108
Protein %	8.5	11	11	17
Phosphorus %	0.27	0.35	0.37	0.30
Calcium %	0.02	0.04	0.06	1.77

1. Propose a model to determine the optimal formulation in order to minimize the feed cost
2. Resolve the model using *Solver* tool from Excel and LINGO.
3. Indicate the formulation and the minimum cost of feed.
4. What are the exact percentages of protein, calcium and phosphorus of the obtained feed?
5. The company needs to buy more barley, but the price has increased by 10 euros/ton. In this situation, should the feed be manufactured with the same raw materials which the optimal solution indicates?
6. The purchasing manager updates market prices and observes that prices of corn and wheat have decreased slightly, exactly 4 euros/ton in both cases. Would this new situation affect production policy and therefore also the purchase of raw materials?
7. The multinational company has launched an environmental program which is a part of the corporate social responsibility, in order to reduce pollution caused by meat production. They are committed to manufacturing the feeds with protein quantities that are closer to animals' requirements. Therefore, they have proposed reducing the maximum percentage of protein to 14.5%. Would this decision affect the formulation and cost of feed? What happens if the proposal is 13%? If possible, indicate the formulation and cost of feed in both cases from sensitivity analysis.
8. Analyse the differences if there are any, between the information given by Excel and LINGO.

CASE STUDY 3: PRODUCTION PROBLEM

A company manufactures three products A, B and C. The three products share four machines M1, M2, M3 and M4 in their production process. For product A we need three operations on machines M1, M3 and M4, for product B only two operations on machines M1 and M3 or on machines M2 and M4 are needed, and product C can be manufactured using machines M1 and M3 or machines M2, M3 and M4.

The time required in minutes per unit produced for each production possibility on each machine, the variable cost of production per minute, the daily capacity of each machine and the minimum daily demands of the three products are presented in the following Table.

The objective consists of determining the production scheme that minimizes the overall variable cost. Solve this problem with LINGO and answer the following questions.

1. How many units of each product are manufactured in each process and what is the overall cost?
2. Which machines have idle capacity and what are these capacity?
3. If it were possible to add an extra time of half an hour per day on machine M1, what effect would it have on the overall production cost?
4. If demand for product B were 40 units, what effect would it have on the overall production cost?
5. What can you say about the effect on the overall cost in the case of an increase in demand for C from 10 to 12 daily units?
6. The company has received an order to produce 5 units per day of a new product, D. Each unit of D requires 2 min on machine M1, 12 min on machine M2 and 6 min on machine M3. The net profit per unit of D is 25 money units. Should this product be manufactured? Justify your answer. In case of an affirmative answer, what would the new value of the objective function be without solving the problem again? Please use the opportunity cost concept. Finally, check if the result indicated is correct.

Table 2.11. Technical and economic data

Product		Process	Time (min/unit)				Minimum daily demand
			M1	M2	M3	M4	
A		1	10		6	3	36
B	B1	1	8		10		45
	B2	2		6		9	
C	C1	1	8		16		10
	C2	2		10	3	8	
Variable cost per min (mu)			40	50	24	30	
Daily capacity in min			480	480	480	480	

CASE STUDY 4: PRODUCTION PROBLEM

A company is planning the apple picking and production of cider for the next season. It manufactures several cider products (natural, extra, brut, black label, etc.) that differ in the mixture of apple varieties, which can be grouped as sweet, sour or bitter. The production process has several phases: pressing, maceration, slow fermentation, clarification and stabilization, bottling and labelling. The process yield is high and produces 0.8 litres cider per kilo of apples.

The company has built a linear programming model to be able to determine how many apples of each variety it should buy to produce 40,000 litres of natural cider and 10,000 litres of extra cider. Table 2 shows the characteristics of the apple varieties and of the cider products. The degree of acidity of the cider must be within the range indicated in the table, as must the sugars in the case of the extra cider. The objective is to minimize the company production costs.

Table 2.12 Characteristics of the apple varieties and types of cider

Characteristics	Apple variety			Cider products	
	1	2	3	Natural	Extra
Degrees of alcohol %	8	6.5	4.2	7	4.8
Volatile acidity gr / litre	1.3	2.1	1.4	1.2 - 2	0.7 – 1.6
Total Acidity gr / litre	2.2	4	3.5	3 – 4.5	3.5 - 4
Sugars gr / litre	70	45	55	-	50 - 60
Price euros/kg.	0.4	0.35	0.3		

```
MODEL:

! Cider production;
! VARIABLES: V1j, V2j y V3j Kg. Apples of variety 1,2 o 3 needed
to produce cider j (N=Natural y E=Extra);

[OBJECTIVE_FUNCTION] MIN=0.4*(V1N + V1E) + 0.35*(V2N + V2E) +
 0.3 *(V3N + V3E);

! Constraints;
[DegreesA_Natural] 0.8*(8*V1N + 6.5*V2N + 4.2*V3N) = 7*40000;
[VolatileA_Min_Natural] 0.8*(1.3*V1N + 2.1*V2N + 1.4*V3N) >= 1.2*40000;
[VolatileA_Max_Natural] 0.8*(1.3*V1N + 2.1*V2N + 1.4*V3N) <= 2*40000;
[TotalA_Min_Natural] 0.8*(2.2*V1N + 4*V2N + 3.5*V3N) >= 3*40000;
[TotalA_Max_Natural] 0.8*(2.2*V1N + 4*V2N + 3.5*V3N) <= 4.5*40000;
[Quantity_Natural] 0.8*(V1N + V2N + V3N) = 40000;

[DegreesA_Extra] 0.8*(8*V1E + 6.5*V2E + 4.2*V3E) = 4.8*10000;
[VolatileA_Min_Extra] 0.8*(1.3*V1E + 2.1*V2E + 1.4*V3E) >= 0.7*10000;
[VolatileA_Max_Extra] 0.8*(1.3*V1E + 2.1*V2E + 1.4*V3E) <= 1.6*10000;
[TotalA_Min_Extra] 0.8*(2.2*V1E + 4*V2E + 3.5*V3E) >= 3.5*10000;
[TotalA_Max_Extra] 0.8*(2.2*V1E + 4*V2E + 3.5*V3E) <= 4*10000;
[Sugars_MinExtra] 0.8*(70*V1E + 45*V2E + 55*V3E) >= 50*10000;
[Sugars_MaxExtra] 0.8*(70*V1E + 45*V2E + 55*V3E) <= 60*10000;
[Quantity_Extra] 0.8*(V1E + V2E + V3E) = 10000;

END
```

```
Global optimal solution found.
  Objective value:                              22246.38
  Infeasibilities:                              0.000000
  Total solver iterations:                            10

                                Variable           Value        Reduced Cost
                                     V1N        16666.67            0.000000
                                     V1E        0.000000           0.1739130E-01
                                     V2N        33333.33            0.000000
                                     V2E        3260.870            0.000000
                                     V3N        0.000000           0.2666667E-01
                                     V3E        9239.130            0.000000

                                     Row    Slack or Surplus      Dual Price
                                      FO        22246.38           -1.000000
                 DegreesA_Natural               0.000000          -0.4166667E-01
                 VolatileA_Min_Natural          25333.33            0.000000
                 VolatileA_Max_Natural          6666.667            0.000000
                 TotalA_Min_ Natural           16000.00            0.000000
                 TotalA_Max_Natural             44000.00            0.000000
                 Quantity_Natural               0.000000          -0.1666667
                 DegreesA_Extra                 0.000000          -0.2717391E-01
                 VolatileA_Min_Extra            8826.087            0.000000
                 VolatileA_Max_Extra            173.9130            0.000000
                 TotalA_Min_Extra               1304.348            0.000000
                 TotalA_Max_Extra               3695.652            0.000000
                Sugars_MinExtra                 23913.04            0.000000
                 Sugars_MaxExtra                76086.96            0.000000
                 Quantity_Extra                 0.000000          -0.2608696

Ranges in which the basis is unchanged:

                                        Objective Coefficient Ranges:

                                  Current        Allowable        Allowable
                      Variable  Coefficient       Increase         Decrease
                           V1N    0.4000000       INFINITY        0.1739130E-01
                           V1E    0.4000000       INFINITY        0.1739130E-01
                           V2N    0.3500000      0.1052632E-01     INFINITY
                           V2E    0.3500000      0.1052632E-01     INFINITY
                           V3N    0.3000000       INFINITY        0.2666667E-01
                           V3E    0.3000000       INFINITY        0.2666667E-01

                                          Righthand Side Ranges:

                                  Current        Allowable        Allowable
                           Row      RHS           Increase         Decrease
        DegreesA_Natural          280000.0        13333.33         12500.00
        VolatileA_Min_Natural     48000.00        25333.33         INFINITY
        VolatileA_Max_Natural     80000.00        INFINITY         6666.667
        TotalA_Min_ Natural       120000.0        16000.00         INFINITY
        TotalA_Max_Natural        180000.0        INFINITY         44000.00
        Quantity_Natural          40000.00        1197.605         1355.932
        DegreesA_Extra            48000.00        571.4286         6000.000
        VolatileA_Min_Extra       7000.000        8826.087         INFINITY
        VolatileA_Max_Extra       16000.00        INFINITY         173.9130
        TotalA_Min_Extra          35000.00        1304.348         INFINITY
        TotalA_Max_Extra          40000.00        INFINITY         3695.652
        Sugars_MinExtra           500000.0        23913.04         INFINITY
        Sugars_MaxExtra           600000.0        INFINITY         76086.96
        Quantity_Extra            10000.00        1038.576         326.4095
```

1. Indicate the minimum cost of production and the quantity of apples of each variety that the company should buy for the next period.

2. What is the value of alcohol in degrees, the volatile and total acidity in the natural cider, produced according to optimal solution of the model?

3. The natural cider is produced using apple varieties 1 and 2. Why is variety 3 not used to make natural cider? Under what conditions which would it be interesting to use variety 3 to produce natural cider?

4. What would the optimal solution be if the price of variety 1 was 0.45 euros/kg? And what would happen if the price was 0.39 euros/kg?

5. Will the optimal solution change if the alcohol degree of natural cider decreases to 6.8 %? What happens if the value is 6%? What the changes and what values remain the same in both cases?

6. What would be the effect on the optimal solution if the maximum level of sugars is 55gr/litre in the extra cider?

CHAPTER 3

GENERAL METHODS OF LINEAR PROGRAMMING

In the previous chapter we graphically solved a linear program with two variables to introduce the basic concepts of linear programming in the most comprehensible and intuitive way possible. However, in practice, problems usually have more variables and constraints and therefore we need some mathematical tools to find the optimal solution. Fortunately, we have a very efficient algebraic technique: the Simplex Method developed by George Dantzig in 1947. The simplex method is an amazing algorithm which received a competitor introduced in 1984 (interior point algorithm of Karmarkar) that seemed to replace it. However, this competitor has not so far fulfilled the expectations that it had initially raised. Today the simplex method remains the basis of software that solves linear programming models in the business sector, both large and small. Only in the case of very large models would the interior point algorithm be preferable and then in combination with the simplex method.

This chapter starts with the definitions of the basic concepts of general linear programming techniques, that is, convex sets, corner-points and basic solutions. We will then explain the Simplex method and the dual phase method as well as techniques with bounded variables.

We will end the chapter with a brief reference to the revised simplex method and the latest developments in linear programming. Professional business managers need to know the basics of the methods of solving linear programming. This knowledge facilitates the formulation of models and the interpretation of the solutions, improving decision making. In addition, the simplex algorithm and its extensions are the basis of sensitivity analysis, as well as other optimization techniques that we will see in following chapters such as integer programming, multiobjective programming and nonlinear programming.

3.1. BASIC CONCEPTS: CORNER-POINTS AND BASIC SOLUTIONS

We will continue using the production model of a power plant as described in the previous chapter. For now, we will simply consider two of the four original constraints. Specifically, the problem will consist of finding the values of the decision variables X_1 and X_2 that

$$(1) \qquad \text{Maximize} \quad 24\,X_1 + 20\,X_2$$

and verify the *constraints*

$$X_1 \geq 0 \text{ and } X_2 \geq 0$$

$$0.5\,X_1 + X_2 \leq 12 \qquad \text{(smoke)}$$

$$1.5\,X_1 + X_2 \leq 24 \qquad \text{(pulverizer)}$$

Figure 3.1 shows the nonnegativity conditions of the variables and constraints for this problem. As we already know, the set of points (X_1, X_2) that meet all constraints form the feasible region.

The *feasible region* of a linear programming model is a *convex set*.

A set is convex when, between two given points, all midpoints also belong to the set. This is a characteristic of the feasible region of any linear programming model and it is the principle for the solution procedure known as simplex algorithm. Another key concept is that of ***corner-points***, which are the vertices of the polygon that forms the feasible region (in the case of two variables).

In the previous chapter, we have also intuitively seen that if a linear programming model has a finite optimal solution, at least one corner-point of the feasible region will be optimal. Figure 3.1 shows that the corner-points of the problem are A, B, C and D. How are these points generated algebraically? First, we formulate the linear programming model *(1)* in the standard form by entering the slack variables X_3 and X_5 to maintain the notation of the original problem.

$$(2) \qquad \begin{aligned} 0.5\, X_1 + X_2 + X_3 \qquad &= 12 \quad (smoke) \\ 1.5\, X_1 + X_2 \qquad + X_5 &= 24 \quad (pulverizer) \end{aligned}$$

Note that now each corner-point of the feasible region can be obtained by setting two variables to zero and solving the resulting set of equations *(2)*. For example, point C, which corresponds to the optimal solution, is obtained by setting X_3 and X_5 to zero. When solving the resulting system of two equations with two unknown quantities we obtain X_1=12 and X_2=6. We can verify this result and obtain the values of the variables for points A, B and D.

However, the selection of variables to be set to zero in order to obtain the corner points is not arbitrary. If we set X_1 and X_5 to zero we obtain the point E (X_2=24 and X_3 = -12). This solution is not feasible as X_3 is negative and therefore not a corner point. However, any feasible solution obtained by this procedure is a corner point and vice versa.

A simple method to determine the values of the variables in the corner points is as follows. If we express the first constraint *(2)* in terms of X_1 and the slack variables, and the second in terms of these latter and X_2 we obtain the following equations

$$(3) \qquad \begin{aligned} &X_1 - X_3 + X_5 = 12 \\ &X_2 + 1.5\, X_3 - 0.5\, X_5 = 6 \end{aligned}$$

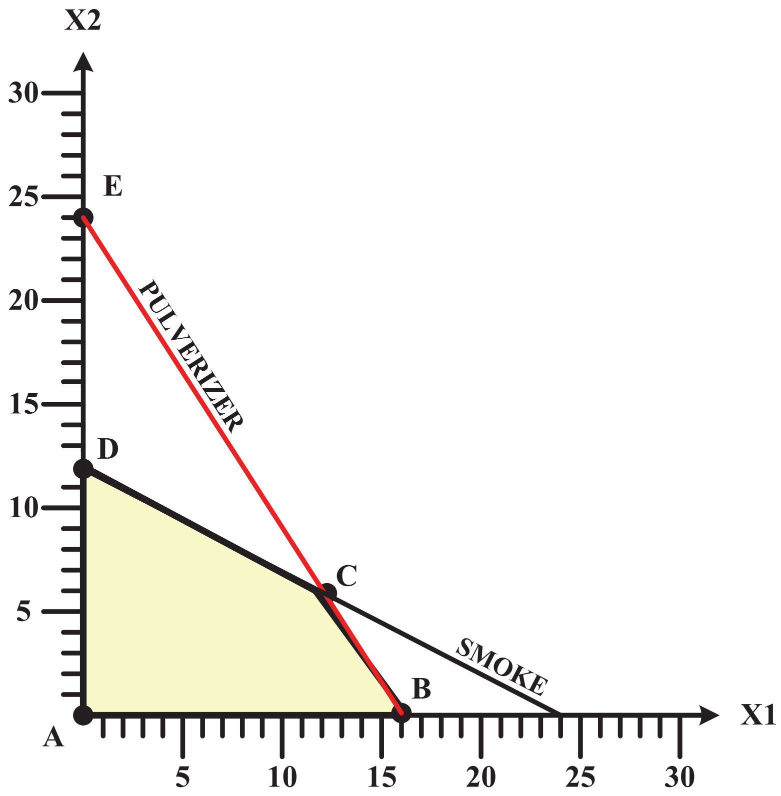


Figure 3.1. Graphic representation

Thus, the first equation of *(3)* is obtained by subtracting the first equation of *(2)* from the second.

When the equations have been transformed in this way, it is known as the canonical form. From the canonical form, the values of X_1 and X_2 can be determined by setting X_3 and X_5 to zero. Likewise, the values of the variables at B (X_1=16 and X_3=4) can be read from the following equations when X_2 and X_5 are zero.

(4) $X_3 + 2/3\ X_2 - 1/3\ X_5 = 4$

$X_1 + 2/3\ X_2 + 2/3\ X_5 = 16$

Note that point A is obtained when setting X_1 and X_2 to zero in equations *(2)*. Therefore, we see that the corner points are related to the algebraic structure of the solution of equations when only a subset of the variables is used.

If we consider a linear programming model in standard form with *n* variables (decision and slack variables) and *m* constraints, we can state that the subset of variables that forms a corner-point is found by setting (*n-m*) variables to zero and solving the resulting system of *m* equations with *m* variables. This system of equations has only one solution.

***Basic solution* of Ax = b is any solution with (*n-m*) variables set to zero.**

Those *(n-m)* variables that are set to zero are the nonbasic variables and the remaining *m* are *the basic variables*.

A *feasible basic solution* is a basic solution in which all the variables are nonnegative.

In a linear programming model with m constraints, two *basic solutions are adjacent* if their sets of basic variables have m-1 in common.

The corner points of the feasible region of a linear programming model are the basic feasible solutions of the equations that represent the constraints of the problem. If a linear programming model has a finite optimal solution, then at least one optimal solution is a corner-point.

The general idea of the simplex method consists of searching through the basic feasible solutions or corner-points until the optimal solution is found. For a standard linear programming model with *n* variables and *m* constraints we will have a basic solution for each group of *n-m* variables that we eliminate. That is, we will have as many as combinations of *n* elements taken from *m* in *m*.

$$\begin{pmatrix} \mathbf{n} \\ \mathbf{m} \end{pmatrix} = \frac{\mathbf{n!}}{\mathbf{(n-m)!m!}}$$

We could think of finding the optimal solution by listing all the feasible basic solutions and choosing the one with the best value of the objective function. However, a linear program with 20 variables, including slack variables, and ten constraints could have up to 184,756 feasible basic solutions. Fortunately, the simplex algorithm is much more efficient and it usually finds an optimal solution after examining between m and 3m basic solutions, where m is the number of constraints. For the previous case, the simplex algorithm would have to evaluate at the most 30 basic solutions.

Concretely, the simplex starts from a basic solution and passes at each iteration of the algorithm to another adjacent basic solution, improving the value of the objective function. When it reaches a basic solution that does not have any adjacent solution that improves the objective function, then that solution is an optimal solution.

Lastly, we would like to emphasize an issue of practical interest that derives from what we have just seen: the optimal solution of a linear programming model is a feasible basic solution. In a model of production planning with *n* activities (not slack ones) and *m* constraints of resources where *n>m* we will never have more than *m* activities with positive levels in the optimal solution. We may have even less if some of the slack variables had positive values. Intuitively, this is not an obvious property and it is difficult to understand if the mathematical solution procedure is not known.

3.2. THE SIMPLEX METHOD

3.2.1. GENERAL CONCEPTS

The general idea of the simplex method consists of starting from a feasible basic solution or corner point and moving to an adjacent feasible basic solution with a better value for the objective function. This process continues until improvements can no longer be made and an optimal solution is therefore found. Thus, the simplex algorithm should consist of:

1. Finding an initial basic feasible solution
2. Finding a basic solution adjacent to the previous one
3. Ensuring that the new basic solution is feasible and, therefore, corner point
4. Ensuring that the new solution is better than the previous one
5. Analysing the simplex method for a maximization problem, that does not imply loss of generality, since any minimization problem can become a maximization problem in the following way

$$\textit{Minimizing } Z = 2\,X_1 + 3\,X_2$$

yields the same solution as

$$\textit{Maximizing } (-Z) = -\,2\,X_1 - 3\,X_2$$

3.2.2. THE SIMPLEX METHOD BY SIMULTANEOUS EQUATIONS

Let us observe equation (2). They represent the corner-point A of Figure 3.1, with X_1 and X_2 set to zero and $X_3 = 12$ and $X_5 = 24$. The slack variables X_3 and X_5 are the basic variables, while the decision variables of the model X_1 and X_2 are the nonbasic variables. This solution is an initial basic solution for the simplex algorithm that consists of not doing anything, that is, not burning any coal. Therefore, the objective function (the quantity of vapor produced) is zero ($Z = 0$). In addition, equations *(2)* are already in canonical form.

We will add the objective function to this set of equations, considering Z as a variable and passing of all the decision variables to the first member. The second member will give us the value of Z with changed sign *(5)*.

$$(5) \qquad \begin{aligned} &0.5\,X_1 + X_2 + X_3 &&= 12 \\ &1.5\,X_1 + X_2 \qquad + X_5 &&= 24 \\ &\mathbf{-\,Z} + 24\,X_1 + 20X_2 &&= \mathbf{0} \end{aligned}$$

Is this solution optimal? To answer this question we will have to see if it is possible to improve the value of the objective function when increasing the value of any of the nonbasic variables. It is only possible to increase the value of Z by increasing the value of the nonbasic variables, since the solution in terms of the basic variables is unique. If we set $X_1 = 1$ the objective function increases by 24 units, while the increase is by 20 if we set $X_2 = 1$. Therefore, corner-points can exist with a better value of Z and point A is not an optimal solution.

We cannot find the optimal corner-point directly, because we do not know which one it is. Therefore, the simplex shifts from one corner-point to another, adjacent, one by improving the objective function. An adjacent corner-point can be reached by replacing one of the basic variables (X_3 or X_5) with one of the nonbasic variables (X_1 or X_2). First, we choose the nonbasic variable that will be the new basic variable, which is the ***entering basic variable***. We then determine which of the basic variables it replaces; that is, which variable is the ***leaving basic variable***. One way of selecting the entering basic variable is to choose the one that provides the highest increase per unit in the objective function. In our case we would choose X_1.

Since each unit of X_1 increases Z by 24 units, we should increase X_1 as much as possible. In fact, the highest value that X_1 can take depends on what happens to the basic variables, because those nonbasic variables will still be zero. From *(5)* we can express X_1 depending on the nonbasic variables, because X_2 is zero, obtaining *(6)*.

$$(6) \qquad \begin{aligned} &X_1 = (12 - X_3)/0.5 \quad o \quad X_3 = 12 - 0.5\,X_1 \\ &X_1 = (24 - X_5)/1.5 \quad o \quad X_5 = 24 - 1.5\,X_1 \end{aligned}$$

Note that in *(6)* as X_1 increases, X_3 and X_5 decrease. Since we allow neither X_3 nor X_5 to become negative, X_1 should not exceed the value that would set one of them to zero first. From *(6)*, the highest value X_1 can take is

$$(7) \qquad X_1 = \min\,(12/0.5 \text{ and } 24/1.5) = 16$$

At this point $X_5 = 0$ and $X_3 = 4$. Therefore X_5 leaves the basis to become a nonbasic variable. The new solution is basic because only two (m) variables are nonnegative and therefore it is also feasible. This solution corresponds to the corner-point B of the Figure 3.1 with X_1=16, X_3=4 and X_2=X_5=0. Note that this point is adjacent to point A.

The next step consists of finding the set of equations in canonical form for this new solution, with X_1 and X_3 as new basic variables and X_2 and X_5 as nonbasic variables. For this, we start from *(5)* and we divide the second equation by 1.5, obtaining the second equation *(8)*. The first equation *(8)* is obtained by taking the first equation of *(5)*, from which we subtract the second equation of *(8)* (which we have just obtained) multiplied by 0.5. The new objective function is obtained by taking the previous objective function and subtracting the second equation of *(8)* multiplied by 24.

$$
\begin{aligned}
(8) \qquad & 2/3\,X_2 + X_3 - 1/3\,X_5 = 4 \\
& X_1 + 2/3\,X_2 + 2/3\,X_5 = 16 \\
& -Z + 4\,X_2 - 16\,X_5 = -384
\end{aligned}
$$

Now you can read the values of the basic variables as $X_1=16$ and $X_3=4$. The new value of Z is 384, which corresponds to the increase of X_1 to 16 with a unitary improvement of 24 (16*24=384). With this, we have finished an iteration of the simplex algorithm.

Now we are dealing again with a basic feasible solution in canonical form, but with a better value for the objective function. We will have to repeat the same steps. Therefore, the second iteration begins by determining whether we can increase the value of the objective function by increasing the value of the nonbasic variables. Note that a unitary increase of X_2 improves the objective function by 4, while a unitary increase of X_5 would cause a reduction of 16 in Z. Thus, it is clear that it is not advisable to increase the value of X_5, because in that case the objective function would decrease in value. So the entering basic variable will be X_2.

Next we will have to determine the leaving basic variable and with the same procedure used in the first iteration we will have

$$
\begin{aligned}
(9) \qquad & X_3 = 4 - 2/3\,X_2 \\
& X_1 = 16 - 2/3\,X_2
\end{aligned}
$$

Thus the highest value that X_2 can take is

$$(10) \quad X_2 = \min\,(4/0.666 \text{ y } 16/0.666) = 6$$

and therefore X_3 is the leaving basic variable. The final operations of iteration 2 consist of finding the canonical form associated with the new basis to determine the values of the basic variables and of Z. The first equation of *(8)* is simply set in canonical form by dividing it by 2/3; the result is the first equation of *(11)*. The second equation of *(11)* is obtained by subtracting the previous one (multiplied by 2/3) from the second equation of *(8)*. Likewise, the new objective function in *(11)* is the objective function of *(8)* from which the first equation of *(11)* multiplied by 4 has been subtracted. The result is the following set of equations.

$$(11) \qquad X_2 + 3/2\, X_3 - 1/2\, X_5 = 6$$

$$X_1 - X_3 + X_5 = 12$$

$$- Z - 6\, X_3 - 14\, X_5 = - 408$$

Note that when we set X_3 and X_5 to zero we obtain $X_1 = 12$, $X_2=6$ and $Z=408$. At this second iteration the objective function has increased by 6*4=24 units.

Now we start the third iteration by analyzing if there are any nonbasic variables that can improve the objective function. We observe that there are none, since X_3 causes a unitary decrease of 6 and X_5 of 14. Therefore, we have found the optimal solution after two iterations.

3.2.3. APPROACHES OF THE SIMPLEX METHOD: ENTERING BASIC VARIABLE AND LEAVING BASIC VARIABLE

We will now formalize the approaches used in the previous section. Once we have an initial basic feasible solution expressed in canonical form, the *entering basic variable* is chosen first. In order to determine this variable, the coefficients of the equation corresponding to the objective function are analyzed and the most favourable one is chosen. As we are maximizing, we choose the highest coefficient because it is the one where Z increases the most. These coefficients are called *reduced coefficients of the objective function or reduced costs.*

The reduced coefficient of a variable X_j is the unitary variation of the objective function per unit of this variable that enters the basis. This variation is the net effect of what the variable that enters basis provides and the variation because the basic variables change its activity level when introducing units of a nonbasic variable.

$$(12) \qquad \textbf{Reduced cost of } X_j\textbf{:} \quad C_j - Z_j = C_j - \Sigma\, \alpha_{ij}\, C_i$$

where C_j is the coefficient of the variable X_j in the objective function. Similarly for C_i, indicating the subindex that is a basic variable located in the equation or row i. The coefficient α_{ij} is the coefficient of variable j in the equation or row i. Let us see an example.

At the second iteration we choose the only one of those nonbasic variables that could improve the objective function as the entering basic variable: X_2. Why is the improvement of Z only 4, if we increase X_2 by one unit when its C_j is 20? In other words, to decide from equations *(8)* to burn a ton of coal B increases the quantity of vapor produced by four thousand pounds. This is the case, because when increasing the value of X_2 by one unit the objective function increases by 20, but X_3 and X_1 decrease by 2/3, which causes a decrease of the objective function of (2/3) C_3 + (2/3) C_1 = (2/3) 24 = 16. Therefore, 20 - 16 = 4 is the net effect due to the changes of levels of activity of the basic variables when a unit of the entering basic variable is increased.

Regarding our problem, we can add that the basic solution corresponding to *(8)* consists of burning 16 tons of coal A, nothing of B, the full capacity of the pulverizer machine has been reached (its slack variable X_5 is zero) and we can still increase the emission of smoke by 4 more units. When considering whether to burn coal B, we have to reduce the quantity of coal A to have the necessary capacity in the pulverizer to burn 1 ton of coal B. As we can see in (5), 1 ton of coal A only requires 3/2 units of the pulverizer and 1 ton of coal B, 1 unit, therefore to pulverize one ton of coal B it is necessary to decrease the burning of coal A by 2/3.

The previous reasoning is based on the use of scarce resources, however, it can be generally applied to any type of variables and constraints. The reduced coefficient of the objective function (C_j - Z_j) always represents the variation of the objective function per entering nonbasic variable unit. If it is positive, it will increase the value of the objective function and if it is negative it will decrease this value. The reduced cost of the basic variables is always zero.

When there are several nonbasic variables in a basic solution that increase Z (if we are maximizing), one approach is to select the entering basic variable with the best unitary improvement. When there is no positive reduced coefficient, we will not be able to improve Z and therefore, we have reached the optimal solution.

CRITERION TO CHOOSE THE ENTERING BASIC VARIABLE

The entering basic variable is the variable with the highest (C_j-Z_j) value (Maximization).

OPTIMALITY CRITERION

A feasible basic solution is optimal if any (C_j-Z_j) ≤ 0 for the nonbasic variables (Maximization).

Previously, we have seen that a minimization problem, Min Z, can be expressed as a maximization Max (-Z) problem. However, the simplex criterion can also be modified conveniently. Thus, the entering basic variable would be the one which has the smallest (C_j-Z_j), and the optimality criteria that (C_j-Z_j) $\geq$ 0.

Let us now look at what happens with the levels of activity of the basic variables when a nonbasic variable enters, for example X_2 at the second iteration. We have just seen how a unitary increment of X_2 causes a decrease of 2/3, in X_3 as well as in X_1 as can be seen in *(8)*. Similarly, an increase of $X_2=\theta$ in *(8)* will cause some proportional decrease of the basic variables. Concretely

$$(13) \qquad X_3 = 4 - \alpha_1 X_2 = 4 - 2/3\, X_2$$

$$X_1 = 16 - \alpha_2 X_2 = 16 - 2/3\, X_2$$

As X_2 is increased, X_1 or X_3 will gradually decrease until one of them is zero. If we continue increasing X_2, one or the other or both would be negative, therefore the solution would become an infeasible solution and we can never let this happen. Therefore, the highest value that X_2 can take, maintaining feasibility, is

$$(14) \qquad \theta = \min\,(4/\,2/3,\ 16/\,2/3) = 6$$

if any of those α_i had been negative or zero, we would have not carried out the corresponding quotient, since the basic variable would increase its value or would remain with the same level of activity. In this case, the basic variable would never be zero and, therefore, it could not be replaced by the entering basic variable.

CRITERION FOR THE LEAVING BASIC VARIABLE

Given α_i of the entering basic variable, the leaving basic variable is the one that satisfies

$$(15) \qquad \theta = \min \frac{(\text{value of the basic variable } X_i)}{\alpha_i} \text{ for any } \alpha_i > 0$$

θ is the value of the new basic variable X_j in the new solution.

If there are no positive α_i, θ can be increased without bound because none of the variables in the current basis will be zero. As θ increases, the value of the objective function also increases. If the entering basic variable does not have an upper bound, the objective function could increase to infinity. This situation is known as *unbounded solution*.

CRITERION FOR UNBOUNDED SOLUTIONS

If for some nonbasic variable with $(C_j-Z_j) > 0$ all the values α_i are not positive, the linear programming model does not have a finite optimal solution.

CHANGE OF BASIS:

When we pass from a basic solution BS^0 to an adjacent basic solution BS^1 the following formulas indicate how the levels of activity of the basic variables and the value of the objective function change:

$$(16) \qquad X_i^1 = X_i^0 - \alpha_{ij}\,\theta_j$$

$$X_j^1 = \theta_j$$

$$\Delta Z = \theta_j\,(C_j - Z_j)$$

The first equation tells us the value of $\mathbf{X_i^1}$ (basic variable i in the basic solution BS^1) when units of the nonbasic variable j (θ_j) enter in the basis. The coefficient of the equation α_{ij} is thus decreasing the basic variable i in the above solution ($\mathbf{X_i^0}$) per unit increase in the nonbasic variable j.

The increase of the objective function $\mathbf{\Delta Z}$ is the number of units of the entering basic variable (θ_j) by its reduced coefficient ($\mathbf{C_j - Z_j}$).

3.2.4. SIMPLEX TABLEAU

The calculations of the simplex method are carried out in a more appropriate way by using a table structure known as *simplex tableau*. Table 3.1 represents the initial simplex tableau corresponding to the corner-point A of Figure 3.1 and equations *(5)*.

In the first column of Table 3.1 we specify the name of the basic variables, then we have a column for each variable of the linear programming model in the standard form, another for the second member of the constraints and finally, a column to specify the quotient between the value of the basic variable and the α_{ij} coefficient of the entering basic variable. Note that the first row of the table, corresponding to X_3 as basic variable, is the first equation of *(5)*, the second row that has X_5 as basic variable is the second equation of *(5)* and, similarly, the third row of C_j-Z_j is the equation of the objective function as expressed in *(5)*. Therefore, it is as if we left Z as a basic variable in all the iterations of the algorithm, but with the sign changed. This is why in the cell of column bi and the row (C_j-Z_j) of the simplex tableau will appear the value of the objective function will appear with the sign changed. Note also that the coefficients of the basic variables make up the unit matrix, therefore, the values of these variables are those shown in column bi.

Applying the entering and leaving criteria of the basic variables, we select first X_1 as entering basic variable, as it produces the highest unitary increase in the objective function. As leaving basic variable, the basic variable chosen is the first to be zero, that is, X_5. Thus, the new basis will be formed by X_3 and X_1. It is important to note that at this precise step the algorithm of the simplex implicitly considers the nonnegativity conditions of the variables.

Table 3.1. Initial simplex tableau

BASIC VAR.	X_1	X_2	X_3	X_5	b_i	b_i/α_i
X_3	0,5	1	1	0	12	24
X_5	1,5	1	0	1	24	16
C_j - Z_j	24	20	0	0	0	

Table 3.2. Second simplex tableau

BASIC VAR.	X_1	X_2	X_3	X_5	b_i	b_i/α_i
X_3	0	2/3	1	-1/3	4	6
X_1	1	2/3	0	2/3	16	24
C_j - Z_j	0	4	0	- 16	- 384	

The following step consists of transforming the first tableau into the canonical form corresponding to the new basis. ***Pivot row*** is the row of the leaving basic variable ***and pivot column***, the column of the entering basic variable. ***Pivot*** is the common element of the pivot column and the pivot row, while the ***semipivots*** are the remaining elements of the column of the entering basic variable.

Table 3.3. Third simplex tableau: optimal solution

BASIC VAR.	X_1	X_2	X_3	X_5	b_i
X_2	0	1	3/2	-1/2	6
X_1	1	0	-1	1	12
C_j - Z_j	0	0	-6	-14	-408

To set the corresponding tableau to the following basic solution in canonical form it is necessary that the pivot element be 1 and all the other elements of the pivot column, zero. This is achieved by applying the following rules when passing from an BS to an adjacent BS[1]:

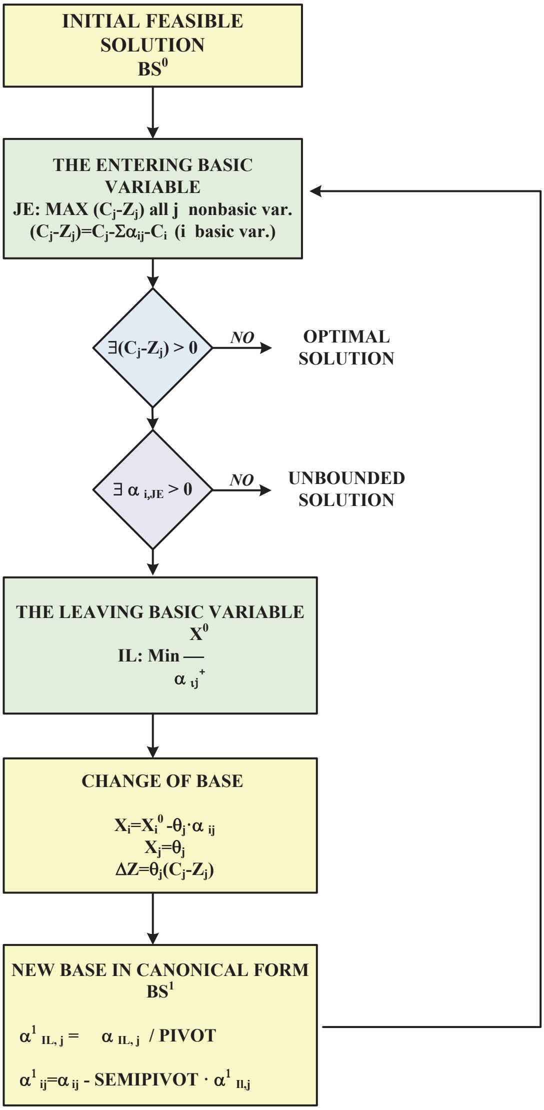
$$\alpha^{1}_{IL,j} = \frac{\alpha_{IL,j}}{\alpha_{IL,JE}} = \frac{\alpha_{IL,j}}{PIVOT}$$

$$\alpha^{1}_{ij} = \alpha_{ij} - SEMIPIVOT \quad \alpha^{1}_{IL,j} \qquad (17)$$

In summary, pivot row $\alpha_{IL,j}$, is transformed by dividing it by the pivot. The remaining lines, α^{1}_{ij}, including (C_j-Z_j), are transformed by subtracting the pivot row obtained in the first place and multiplied by the corresponding semipivot number. Chart 1 shows an outline of the steps of the simplex algorithm.

Table 3.3 is the simplex tableau of the optimal solution. As we already knew and as we can see in this table, the optimal solution consists of using 12 ton/h of coal A and 6 ton/h of coal B, obtaining a vapor production of 408 thousand pounds. Furthermore, this table provides additional information such as the opportunity costs of the scarce resources. What is the dual price of the constraint of smoke? What is its practical meaning?

CHART 3.1. SIMPLEX ALGORITM (MAXIMIZATION)



3.3. INITIAL BASIC FEASIBLE SOLUTION AND ARTIFICIAL VARIABLES. THE TWO-PHASE METHOD.

In the previous example we obtained the basic initial solution from the slack variables, but many problems do not have an initial solution in canonical form. If there are equality constraints, higher or equal or second non positive members (=, ≥ or $b_i \leq 0$), the only additional problem outlined is to *identify the initial basic feasible solution.* In this case, there is no guarantee that a feasible solution exists. Therefore, a systematic and efficient procedure is required to generate this basic initial and feasible solution, if it exists.

One of the advantages of the simplex method is that the method itself is able to generate its own initial basic feasible solution, provided that one exists. If it does not exist, the simplex method specifies that the problem does not have a solution.

When we have equality constraints in a model, there is no slack variable for this constraint that allows us to obtain the initial basic feasible solution. We could consider substituting the equality constraint for other two, one with ≤ and the other with ≥. This alternative is not very advisable because it increases the number of constraints of the problem. The technique of the artificial variable is preferable. This consists of adding, to the left side of the constraint, a fictitious variable that will be called *artificial variable* and whose only mission is to be the basic variable of that constraint in the initial basic solution.

In the case of ≥ constraints, the slack variable enters to be subtracted, and if we multiplied the equation by (-1), we would find the second member negative and therefore with an initial infeasible solution. Therefore, in this case we also add an *artificial variable*.

In summary, when we have **problems with = or ≥ constraints**, an **artificial variable** is added for each one of the constraints of this type. This new problem is called **the augmented problem**. From the artificial variables we obtain an initial basic solution that is infeasible for the original problem because the artificial variables do not have a meaning in it. Therefore, the first thing to do is to find an initial basic feasible solution for our real problem, removing all the artificial variables from the basis. From this new basic solution we will continue searching for the optimal solution.

We will see how the **two-phase method** allows us to **find the initial basic feasible solution**. This method will be explained with the energy generation example reformulated as a model of minimization of production costs. The additional constraint of having to produce a minimum of 216 thousand pounds of vapor is added.

(18) $\quad$ *Minimize* $24 X_1 + 15 X_2$

$$X_1 \geq 0 \text{ and } X_2 \geq 0$$

$$0.5 X_1 + X_2 \leq 12$$

$$1.5 X_1 + X_2 \leq 24$$

$$24 X_1 + 20 X_2 \geq 216$$

In the process that we follow to find the optimal solution we can distinguish two phases. The first corresponds to the iterations required until a feasible solution is found, without the artificial variables. The second phase consists of the additional iterations performed once a feasible basic solution has been found until the optimal solution is reached.

The Two-phase method uses two different objective functions, one in each phase. The algorithm starts by introducing the artificial variables that are required to obtain an initial basic solution.

Phase 1:

The simplex method is used to solve the augmented linear programming whose objective function is the following

(20) ***Minimize Z = Sum of all of the artificial variables***

and the constraints are the original constraints of the model plus the artificial variables and nonnegativity constraints for all decision and artificial variables. The optimal solution obtained for this problem (Z=0) will be a feasible basic solution for the original problem.

Phase 2:

The artificial variables are removed, since their value is zero and the simplex is applied from the feasible basic solution obtained after phase 1 until the optimal solution is found. The objective function of this phase is that of the real problem (minimizing the production cost).

(21) ***Minimize Z = Objective function of the problem =*** $\boldsymbol{24\,X_1+15X_2}$

Table 3.4 represents the initial simplex tableau for phase 1 of the example of cost minimization in energy production. Note that row C_j-Z_j should be calculated from *(12)*. That is

Reduced cost of nonbasic variable $\boldsymbol{X_j}$***:*** $\boldsymbol{C_j - Z_j = C_j - \Sigma\, \alpha_{ij}\, C_i}$

where C_j is the coefficient of the nonbasic variable in the objective function (20). C_i is the coefficient of the basic variable i in the objective function (20) and $\boldsymbol{\alpha_{ij}}$ is the coefficient of variable X_j in the row i. For example,

$C_1\text{-}Z_1 = C_1 - (\boldsymbol{\alpha_{11}}\, C_3 + \boldsymbol{\alpha_{21}}\, C_5 + \boldsymbol{\alpha_{31}}\, C_{\text{Artificial}}) = 0 - (0.5*0 + 1.5*0 + 24*1) = -24$

Table 3.4. PHASE 1: Initial simplex tableau

BASIC VAR.	X_1	X_2	X_3	X_5	X_6	X_a	b_i	b_i/α_i
X_3	0.5	1	1	0	0	0	12	24
X_5	1.5	1	0	1	0	0	24	16
X_a	24	20	0	0	- 1	1	216	9
C_j - Z_j	-24	-20	0	0	1	0	- 216	

Table 3.5. PHASE 1: Second simplex tableau

BASIC VAR.	X_1	X_2	X_3	X_5	X_6	X_a	b_i
X_3	0	7/12	1	0	1/48	-1/48	7.5
X_5	0	-1/4	0	1	1/16	-1/16	10.5
X_1	1	5/6	0	0	- 1/24	1/24	9
C_j - Z_j	0	0	0	0	0	1	0

From Table 3.4 the logic of the simplex means that X_1 enters in the basis because it has the smallest (C_j-Z_j) and we are minimizing the sum of the artificial variables. As there is only one, we have to set it to zero. When you enter X_1 all basic variables decrease in value. For example, X_3 decreases by 0.5 for each X_1 unit that enters and X_a decreases by 24. The first basic variable that reaches zero exits the basis, being the leaving basic variable, in this case it is the artificial variable. Therefore, in one iteration we finish phase 1 whose optimal solution is shown in Table 3.5.

Note that Table 3.6, initial chart of phase 2, is nothing more than the final tableau of phase 1 except that we have removed the column of the artificial variable and recalculated the C_j-Z_j row. It is logical that it should be this way because the objective function is now different. Specifically the objective function of the problem, i.e. minimize the cost of production (18). By applying the two-phase method it may not be possible to find a feasible solution to the original problem. However, the technique of the artificial variables indicates that we have a situation of this type. In this example, Table 3.7 corresponds to the optimal solution because no nonbasic variable can reduce the cost, i.e. decrease the value of the objective function. In this case, the optimality criterion is that reduced costs are greater than or equal to zero.

Table 3.6. PHASE 2: Initial simplex tableau

BASIC VAR.	X1	X2	X3	X5	X6	b_i	b_i/α_i
X3	0	7/12	1	0	1/48	7.5	12.86
X5	0	-1/4	0	1	1/16	10.5	
X1	1	5/6	0	0	- 1/24	9	10.8
$C_j - Z_j$	0	-5	0	0	1	- 216	

Table 3.7. PHASE 2: Second simplex tableau: optimal solution

BASIC VAR.	X1	X2	X3	X5	X6	b_i
X3	-0.7	0	1	0	0.05	1.2
X5	0.3	0	0	1	0.05	13.2
X2	6/5	1	0	0	-0.05	10.8
$C_j - Z_j$	6	0	0	0	0.75	- 162

If the original problem does not have feasible solutions, then any optimal solution obtained in phase 1 of the Two-phase method leads to a final solution that contains at least one artificial variable higher than zero. Otherwise, all of them are equal to zero.

OPTIMALITY CRITERION

The optimal solution for the augmented problem is also optimal for the original problem if there are no artificial variables with non-zero values.

If not all of the artificial variables are eliminated from the basis, the original problem does not have a feasible solution.

3.4. SIMPLEX ALGORITHM WITH BOUNDED VARIABLES

In many real problems the decision variables of the model are lower or upper bounded. For example, in problems of production planning the amount to manufacture can be restricted by minimum demands to cover for some clients or for maximum demands according to market studies. We might think that by increasing the number of constraints of the model we will complicate the resolution, increasing the number of iterations necessary to reach the optimal solution. However, the consideration of this type of constraints does not necessarily involve increasing the matrix of technical coefficients of the problem. These conditions are considered in a special way, so that the efficiency of the algorithm increases, as we will see in the following example. Because of this, optimization software usually recommends that users bound the variables of the model, if possible, although they might not have real bounds. Let us look at the operation of the simplex technique with bounded variables.

3.4.1. LOWER BOUND TECHNIQUE

One of the best known applications of linear programming is the allocation of limited production resources. We will see an example of this type that will allow us to demonstrate its use in the identification of *bottlenecks* that a company may have and will also explain the lower bound technique. Later, we will also use this example to explain the foundations of the dual price and the dual simplex algorithm.

Suppose that a company manufactures three products A, B and C that must be processed through five departments. Table 3.8 displays the data for the problem. The objective is to find out the weekly production of A, B and C that maximizes profits.

To formulate the model, we first define the nonnegative **variables**:

$A \geq 0$ *manufactured units of product A per week*

$B \geq 0$ *manufactured units of product B per week*

$C \geq 0$ *manufactured units of product C per week*

The **objective function** will be to maximizing the total profit

$$MAX\ 20\ A + 18\ B + 21\ C$$

Table 3.8. Production problem by departments

Product	Minimum demand	Unitary benefit	Number of processing hours per product unit				
			Dep1	Dep2	Dep3	Dep4	Dep5
A	100	20	0.2	0.50	0.1	0.02	0.05
B	180	18	0.1	-	0.3	0.02	0.06
C	75	21	0.3	0.07	0.1	0.02	0.05
Capacity of the departments (weekly hours)			160	80	80	40	40

The **constraints** of the model refer to the minimum demands and the capacities of the different departments:

$$
\begin{aligned}
(21) \qquad & DemA: && A \geq 100 \\
& DemB: && B \geq 180 \\
& DemC: && C \geq 75 \\
& Dept1: && 0.2\ A + 0.1\ B + 0.3\ C \leq 160 \\
& Dept2: && 0.5\ A + 0.07\ C \leq 80 \\
& Dept3: && 0.1\ A + 0.3\ B + 0.1\ C \leq 80 \\
& Dept4: && 0.02\ A + 0.02\ B + 0.02\ C \leq 40 \\
& Dept5: && 0.05\ A + 0.06\ B + 0.05\ C \leq 40
\end{aligned}
$$

The last two constraints are redundant and we could remove them, but we will keep them for pedagogical purposes.

The optimization software does not include the variable bounds as constraints and encourages the user to identify them as special constraints to use a more efficient algorithm. How does it solve a problem with lower bound variables? The simplex method uses an implicit lower bound of zero for all the variables. This is achieved by setting all the nonbasic variables to zero and all the basic ones to be nonnegative. The nonnegativity of basic variables is maintained by the rule of choosing the leaving basic variable.

When we have an X_j variable with an L_j lower bound, we should enter a constraint $X_j \geq L_j$. We can avoid it by defining a new variable Y_j that represents the amount by which X_j exceeds its lower bound:

$$(22) \qquad Y_j = X_j - L_j \quad \text{being } Y_j \geq 0$$

Y_j substitutes X_j in all of the constraints and in the objective function. This change of variable modifies the right-hand side of the constraints from bi to $(bi - a_{ij} L_j)$ and gives the objective function an initial value of $Z^0 = C_j L_j$.

The linear programming software automatically carries out these transformations and it undoes them again before showing the optimal solution. We will check this procedure in LINGO.

For a better understanding of this technique we will apply it to our previous example. We define,

$$(23) \qquad \begin{aligned} Ya &= A - 100 \quad & Ya \geq 0 \\ Yb &= B - 180 \quad & Yb \geq 0 \\ Yc &= C - 75 \quad & Yc \geq 0 \end{aligned}$$

the problem being as follows

$$(24) \qquad \begin{aligned} &Max\ 20\ (Ya + \mathbf{100}) + 18\ (Yb + \mathbf{180}) + 21\ (Yc + \mathbf{75}) \\ &0.2\ (Ya + \mathbf{100}) + 0.1\ (Yb + \mathbf{180}) + 0.3\ (Yc + \mathbf{75}) && \leq 160 \\ &0.5\ (Ya + \mathbf{100}) + 0.07\ (Yc + \mathbf{75}) && \leq 80 \\ &0.1\ (Ya + \mathbf{100}) + 0.3\ (Yb + \mathbf{180}) + 0.1\ (Yc + \mathbf{75}) && \leq 80 \\ &0.02\ (Ya + \mathbf{100}) + 0.02\ (Yb + \mathbf{180}) + 0.02\ (Yc + \mathbf{75}) && \leq 40 \\ &0.05\ (Ya + \mathbf{100}) + 0.06\ (Yb + \mathbf{180}) + 0.05\ (Yc + \mathbf{75}) && \leq 40 \end{aligned}$$

and simplifying

$$(25) \qquad \begin{aligned} &Max\ \ \mathbf{6815} + 20\ Ya + 18\ Yb + 21\ Yc \\ &0.2\ Ya + 0.1\ Yb + 0.3\ Yc && \leq \mathbf{99.5} \\ &0.5\ Ya + 0.07\ Yc && \leq \mathbf{24.75} \\ &0.1\ Ya + 0.3\ Yb + 0.1\ Yc && \leq \mathbf{8.5} \\ &0.02\ Ya + 0.02\ Yb + 0.02\ Yc && \leq \mathbf{32.9} \\ &0.05\ Ya + 0.06\ Yb + 0.05\ Yc && \leq \mathbf{20.45} \end{aligned}$$

By introducing the slack variables X_1, X_2, X_3, X_4 and X_5, passing inequalities to equalities and applying the simplex algorithm we obtain Tables 3.9 and 3.10. The initial table corresponds to the initial basic feasible solution that represents the production of the minimum demands required for the three products. This solution has a value of Z = 6815, as obtained in (24) and (25) (see bold numbers).

At the first iteration the entering basic variable is the variable that most increases the objective function per nonbasic variable unit, in this case, Yc. After carrying out the basis change and obtaining the second tableau we see that this corresponds to an optimal solution, because any modification in a nonbasic variable would worsen the value of the benefit (all the Cj-Zj ≤ 0).

Therefore the optimal solution is the following:

$$
(26) \qquad \begin{aligned}
A &= Ya + 100 = 100 \\
B &= Yb + 180 = 180 \\
C &= Yc + 75 = 160 \\
Benefit &= Z = 8600
\end{aligned}
$$

In the tableau corresponding to the optimal solution of this problem, we can see that department 3 is the **bottleneck** of the company, since it is limiting its productive capacity, while the other four departments maintain idle resources. This example shows the utility of linear programming in production planning problems for identifying bottlenecks.

Table 3.9. Initial simplex tableau

BASIC VARIABLE	Ya	Yb	Yc	X_1	X_2	X_3	X_4	X_5	b_i	b_i/α_{ij}
X_1	0.20	0.10	0.30	1	0	0	0	0	99.50	331.6
X_2	0.50	0.00	0.07	0	1	0	0	0	24.75	353.6
X_3	0.10	0.30	0.10	0	0	1	0	0	8.50	85
X_4	0.02	0.02	0.02	0	0	0	1	0	32.90	1645
X_5	0.05	0.06	0.05	0	0	0	0	1	20.45	409
C_j-Z_j	20	18	21	0	0	0	0	0	- 6815	

Table 3.10. Second simplex tableau: optimal solution

BASIC VARIABLE	Ya	Yb	Yc	X_1	X_2	X_3	X_4	X_5	b_i
X_1	-0.10	-0.8	0	1	0	-3	0	0	74
X_2	0.43	-0.21	0	0	1	-0.7	0	0	18.8
Yc	1	3	1	0	0	10	0	0	85
X_4	0	-0.04	0	0	0	-0.2	1	0	31.2
X_5	0	-0.09	0	0	0	-0.5	0	1	16.2
C_j-Z_j	-1	-45	0	0	0	-210	0	0	-8600

Is the company using its resources appropriately? How could it improve? What would happen if the demand for A or B increased? And which would be preferable?

3.4.2. UPPER BOUND TECHNIQUE

When the variables have upper bound constraints, a simple transformation of the variables is not enough for considering these restrictions. Some modification of the rules of the simplex method are also necessary.

If the variable X_j has an upper bound with the value U_j

$$(27) \qquad \boldsymbol{X_j \leq U_j}$$

we define a new $\boldsymbol{X_j'}$ variable so that $\boldsymbol{X_j + X_j' = U_j}$

$\boldsymbol{X_j'}$ is the complementary variable of $\boldsymbol{X_j}$.

1) If X_j is at its upper bound U_j, then X_j' will be at its lower bound and vice versa.

2) The technical coefficients of X_j' are the negative of X_j.

The upper bound routine uses these two properties to eliminate the upper bound constraints. We will see how X_j is used as a nonbasic variable when $X_j=0$ and X_j' is used as a nonbasic variable when X_j is at its upper bound. If the value of the variable is between the two limits, X_j or X_j' both can be basic.

Let us solve the following model

$$(28) \qquad \begin{aligned} &Max\ 4X_1 + 5X_2 \\ &2X_1 + 3X_2 \leq 9 \\ &2X_1 + X_2 \leq 9 \\ &\mathbf{1 \leq X_1 \leq 4} \\ &\mathbf{0 \leq X_2 \leq 1} \end{aligned}$$

Both variables of the model have upper bound constraints and in addition, one of them has a lower bound higher than zero. Therefore, we will begin by defining a new variable Y_1 that expresses X_1 value that is above its lower bound 1.

$$(29) \qquad \mathbf{Y_1 = X_1 - 1} \ \Rightarrow X_1 = Y_1 + 1$$

And we will also define a complementary variable Y_1' which is the difference between the value of Y_1 and its upper bound, which is 3. Similarly X_2' is defined in association to the variable X_2. The sum of both must be the upper bound, in this case 1.

$$(30) \qquad \begin{aligned} &\mathbf{Y_1 + Y_1' = 3} \\ &\mathbf{X_2 + X_2' = 1} \end{aligned}$$

Table 3.11 shows the initial tableau. Previously we have substituted X_1 for Y_1+1 in the equations of the model (28). Therefore, in this table Y_1 appears instead of X_1. Table 3.11 corresponds to an initial basic feasible solution where $\mathbf{Y_1}$ is the nonbasic variable and therefore $\mathbf{X_3 = 7}$**,** $\mathbf{X_4 = 7}$ and $\mathbf{Z=4}$, when $X_1 = Y_1+1$ in (28). That is, we start with a solution where $X_1=1$, which is equivalent to $Y_1= 0$.

The entering basic variable at the first iteration is the variable which has the highest reduced cost C_j-Z_j that is X_2. Applying the simplex method, X_3 would be the leaving basic variable, because it is the first basic variable that reaches the zero value. However, 2.33 units of X_2 are necessary to be set to zero X_3 and they cannot be introduced because X_2 is upper bounded with the value 1. As **X_2 reached its upper bound before any basic variable reaches zero**, a variable change is carried out. X_2 is replaced by X_2' so that the only necessary changes in the tableau to obtain the next tableau, table 3.12, consist of upgrading the column bi with the formula of the basis change (16) and changing the sign to the coefficients of column X_2'.

From table 3.12 we choose variable Y_1 to be the entering basic variable because is the only one that increases the value of Z, and X_3 would leave the basis, because is the first basic variable to reach zero. The table corresponding to the following basic solution, Table 3.13, is calculated by applying the simplex standard rules, that is, with the pivot and semi pivot formulae (17).

For the basic solution to table 3.13, the only variable which increases the objective function is X_2' and, by increasing its value, Y_1 increases by 3/2 and X_4 decreases by 2 for each additional unit in X_2'. Thus X_4 can reach zero and Y_1 can reach its upper bound. The latter is what happens first and the table corresponding to the next feasible basic solution is a little more complicated than in previous cases as, in addition to applying the formulae to change the basis (17), other considerations must be taken into account.

In table 3.14 the value of the entering basic variable $\mathbf{X_2'}$ is 2/3. This value is not obtained by dividing b_i by the α_i coefficient. In particular, 2/3 is what has to enter in the basis of $\mathbf{X_2'}$ for the basic variable Y_1 to reach its upper bound of 3. Moreover as the basic variable Y_1 reaches its upper bound, it is substituted by Y_1'as a nonbasic variable. Therefore, the coefficients of column $Y_1'$are multiplied by (-1).

Table 3.11. Initial simplex tableau

BASIC VAR.	Y_1	X_2	X_3	X_4	b_i	b_i/α_i
X_3	2	3	1	0	7	7/3=2.33
X_4	2	1	0	1	7	7/1=7
$C_j - Z_j$	4	5	0	0	- 4	

Table 3.12. Simplex tableau: first iteration

BASIC VAR.	Y_1	X_2'	X_3	X_4	b_i	b_i/α_i
X_3	2	- 3	1	0	4	4/2=2
X_4	2	- 1	0	1	6	6/2=3
$C_j - Z_j$	4	- 5	0	0	- 9	

Table 3.13. Simplex tableau: second iteration

BASIC VAR.	Y_1	X_2'	X_3	X_4	b_i	b_i/α_i
Y_1	1	-3/2	1/2	0	2	1/3/2=2/3
X_4	0	2	- 1	1	2	2/2=1
$C_j - Z_j$	0	1	- 2	0	- 17	

Table 3.14. Simplex tableau third iteration: optimum solution

BASIC VAR.	Y_1'	X_2'	X_3	X_4	bi
X_2'	2/3	1	- 1/3	0	2/3
X_4	- 4/3	0	- 1/3	1	2/3
$C_j - Z_j$	- 2/3	0	- 5/3	0	- 17.66

From Table 3.14 and taking into consideration (29) and (30) the optimal solution is

$$X_1 = 4$$
$$X_2 = 0.33$$
$$Z = 17.66$$

In brief, the upper bound technique applies the following criteria:

VALUE OF THE INCREASING NONBASIC VARIABLE

$$(31) \qquad X_j = \theta_j = \min \; [\beta, U_j, \delta_j]$$

$$\beta = \min \frac{X i}{\alpha_i} \; \forall \alpha_i > 0$$

U_j = upper bound of the entering basic variable

$$\delta = MIN \frac{\text{base value } Xi \text{ - upper bound } Xi}{\alpha_i} \quad \forall \alpha_i < 0$$

Therefore, the basis change can take place due to three different situations:

1. **β** **The entering basic variable substitutes the leaving basic variable** which reaches zero as in the standard simplex algorithm.

2. ***U_j*** **The entering basic variable** reaches its **upper bound** before a basic variable reaches zero or its upper bound. In this case, **X_j has to be replaced by $X_j´$ or vice versa**. The corresponding column must be multiplied by (-1) to obtain the following simplex tableau, in addition to upgrading the values of the basic variables (b_i) and of the objective function. The new values of the variables are

(32) $X_i^1 = X_i - U_j \alpha_i$

$X_j = U_j \Rightarrow X_j' = 0$

3. **δ** When increasing the entering basic variable, **one of the basic variables reaches its upper bound.** For this reason, in the following table the complementary variable X_k' will appear instead of X_k or vice versa and its column will be multiplied by (-1). The nonbasic variable which is increased, X_j substitutes the basic variable X_k. The new values of the variables are

(33) $X_k = U_j \Rightarrow X_k' = 0$

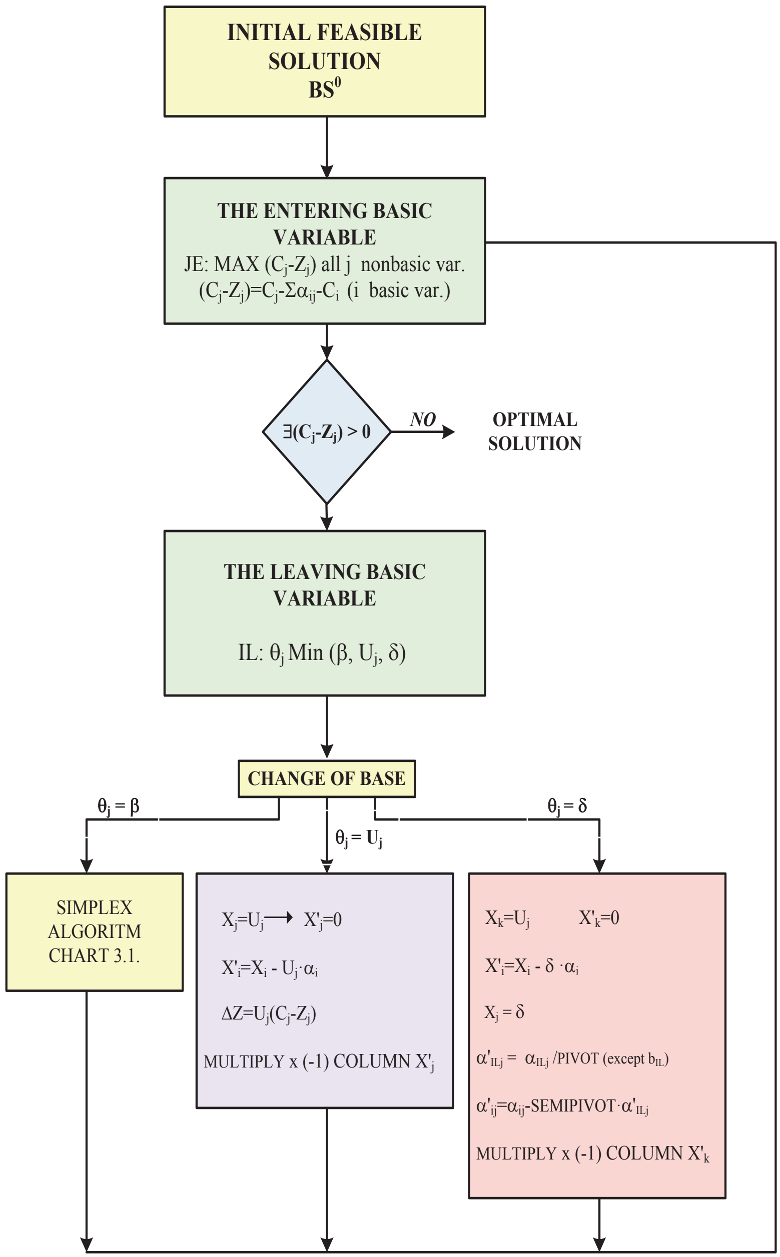
$X_i^1 = X_i - \delta \alpha_i$

$X_j = \delta$

In this case, in order to obtain the new table of the basic solution, it is necessary to apply the formulae of the simplex method, except when calculating the new value of the entering basic variable (δ) and change the variable, and therefore the sign of its corresponding column.

Chart 3.2 outlines the steps of the simplex algorithm with upper bound constraints. Note that the only difference with the simplex method is in the rule for selecting the leaving basic variable. In the simplex method we choose the one which first reaches zero as the leaving basic variable, to avoid an infeasible solution due to a negative variable. In the upper bound technique we select the variable which first becomes infeasible, due to a negative value of the basic variable, exceeding the upper bound of the entering basic variable or the leaving basic variable.

CHART 3.2. UPPER BOUND TECHNIQUE (MAXIMIZATION)



3.5. THE REVISED SIMPLEX METHOD, THE INTERIOR POINT ALGORITM AND THE OPTIMIZATION SOFTWARE

If the simplex algorithm in its form of complete tableau is analyzed in detail, as we have described in sections 3.2 onwards, we will realize that in problems with many more variables than constraints, we are upgrading data at each iteration that are in fact useless. Concretely the only nonbasic variable of interest at each step is the entering basic variable.

The revised simplex method maintains the same fundamental logic as the described simplex method and is nothing other than a simplified **version** that **only calculates and stores the required data** at each moment. It is therefore a more efficient implementation and the one which is in fact used by optimization software.

The revised simplex method has also another important aspect in addition to the advantages of requiring less storage and often less calculation. The computer can have significant **rounding problems** after performing many iterations. This affects the values C_j-Z_j and the ratio of the leaving basic variable. We could, therefore, pick a wrong variable to enter or leave the basis. We reduce these rounding errors to within reasonable limits by determining the inverse of the matrix forming the basis vectors in any iteration. Professional software typically inverts this matrix every few iterations and when the optimality test is met. Special procedures are used to store and update the matrix formed by the coefficients of the basic variables or its inverse, which take into account the dispersion of the matrices helping to correct errors, streamline inversions and reduce storage requirements.

The importance of linear programming at the moment is due to the existence of an extraordinarily efficient algorithm: *the simplex algorithm*, developed by George Dantzig, and the availability of computers that can carry out the large amount of necessary calculations. The theoretical properties of the algorithms can be evaluated through the *computational complexity* and counterexamples have been created to demonstrate that the simplex algorithm is not *polynomial*, but rather it is an *exponential* algorithm. Although in practice the simplex performs very well, researchers have continued to look for polynomial algorithms to solve linear programming models.

Narendra Karmarkar, from the ATT Company, published an article in 1984 in which he announced a new algorithm to solve large linear programming models that had the property of being polynomial. Firstly, he claimed that the algorithm could solve large models up to 50 times faster than the simplex method. As no details of the algorithm were given, for copyright reasons, it could not be verified if this claim was true. Later on, some details were made public and, meanwhile, other researchers have developed applications of the algorithm that, so far, have produced contradictory results.

At the moment, it is not clear which of the two algorithms is more efficient, although it seems that in the future both will be complementary in linear programming. The biggest advantage of the interior-point algorithm is that the increase in the required computational time becomes greater at a smaller rate than the simplex when the size of the problem increases. On the other hand, the high preparation time of Karmarkar's method prevents it from becoming a strong competitor when dealing with relatively small models (dozens or hundreds of functional constraints).

In addition, as we will see in the following chapter, the simplex method is excellent for carrying out the postoptimality analysis, while the interior-point algorithm does not enable one to carry out this analysis efficiently, although it obtains dual prices. In summary, in the future it is foreseen that the simplex will continue to be used as a standard linear programming method and Karmarkar's or one of its versions for very large problems. As the interior-point algorithm converges on the best solution, it is possible that a feasible solution close to the optimal becomes an initial solution for the simplex algorithm, which would allow us to find the optimal solution and carry out the sensitivity analysis.

Next we will demonstrate with an example the approach of the interior-point algorithm. The main concepts of the algorithm are the following.

Concept 1: To obtain a feasible solution that leads to the optimal solution from the interior of the feasible region.

Concept 2: To move in the direction that improves the value of the objective function as fast as possible.

Concept 3: To transform the feasible region in order to place the current trial solution near the center thus allowing a large improvement when concept 2 is carried out.

We will see the previous ideas with the following linear programming model and its graphical representation:

$$\text{Max } Z = X_1 + 2\,X_2$$

$$X_1 + X_2 \leq 8$$

$$X_1 \geq 0 \qquad X_2 \geq 0$$

The algorithm starts with an initial trial solution that should be in the *interior* of the feasible region as in all of the following. Thus, the initial solution should not be in any of the three straight lines that form the boundary of the feasible region ($X_1 + X_2 = 8$, $X_1 = 0$, $X_2 = 0$). $(X_1, X_2) = (2, 2)$ are randomly chosen as an initial trial solution. Next, we have to move in the direction that improves the value of the objective function as fast as possible. This direction is that of the *gradient* of the objective function and it is given by the vector of partial derivatives, that is (1, 2). Note that the components of this vector are the coefficients of the objective function, as this is a linear function.

The algorithm begins with the linear programming model in the standard form

$$\text{Max } Z = X_1 + 2\,X_2$$
$$X_1 + X_2 + X_3 = 8$$
$$X_1 \geq 0 \quad X_2 \geq 0 \quad X_3 \geq 0$$

and the matrix

$$\text{Max} \quad \mathbf{Z} = \mathbf{C}^{\mathbf{T}}\,\mathbf{X}$$
$$\mathbf{A}\,\mathbf{X} = \mathbf{b}$$
$$\mathbf{X} \geq \mathbf{0}$$

Figure 3.2 shows the feasible region of the problem, as well as the path that the trial solutions follow until finding the optimal solution (Hillier and Lieberman, 2010).

In this example we have seen that the interior-point algorithm requires more iterations and more calculations than those performed by the simplex algorithm and in the end it only obtains an approximation to the optimal solution. However, we should take into consideration that this algorithm is designed for large problems with many thousands of functional constraints. In this case the simplex would carry out thousands of iterations, while Karmarkar's algorithm would need a lot less, although with more work per iteration. Currently, the optimization software LINGO incorporates this method (Barrier Solver), in addition to the simplex algorithm.

In this section we have seen that the linear programming software does not use the full tableau simplex method. The revised simplex method is used because of the lower storage and computation effort. Furthermore, the revised simplex method also allows the periodic avoidance of rounding errors accumulated over many iterations. The accuracy of the final solution depends on the tolerances of the program or is specified by the user. Errors are possible due to the way in which the computer manipulates and stores decimal numbers. Thus, small differences from zero, for example 10^{-6}, are considered equal to zero. These limits are called **tolerances**. The choice is not easy, as the small tolerances are difficult to meet and can promote errors. The large tolerances can treat many non-zero data as if they were zero.

The accuracy of the solution can be improved by appropriately **scaled** input data. Whenever possible, very large and very small data in the same model should be avoided, because that increases the risk of accumulation of errors. The software may have options to scale the data of the problem and eliminate internal scaling before presenting the results.

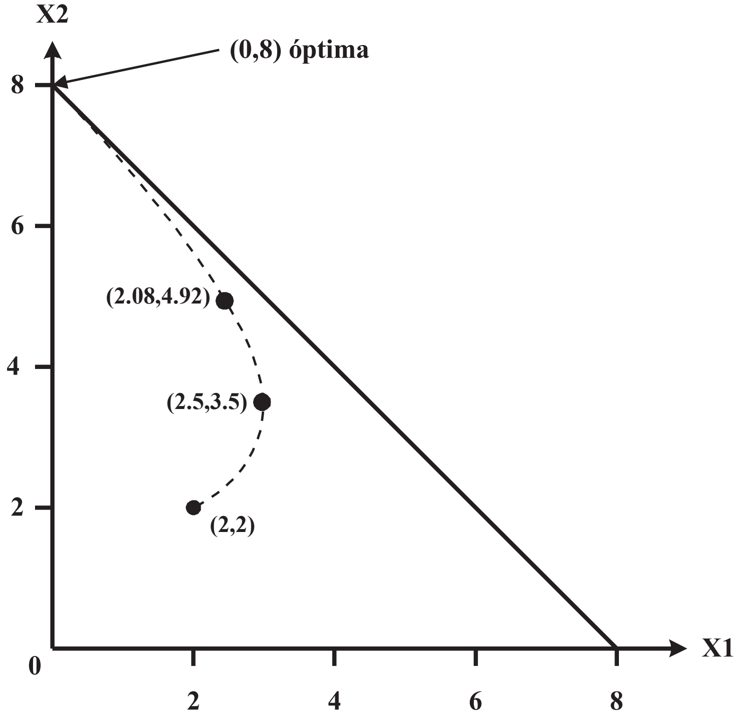


Figure 3.2. Path of the interior-point algorithm

3.6. SUMMARY

In this chapter we have studied general methods for solving any model that adapts to the structure of linear programming as defined in chapter 2. First, we have defined the basic concepts of the solution algorithm *par excellence* of Operations Research/Management Science, the **simplex method**. The feasible region of a linear programming model is a convex set and if it has a finite optimal solution, at least one optimal solution is a corner-point (**feasible basic solution**).

The simplex method starts from a feasible basic solution and it moves to another adjacent feasible basic solution and improves the value of the objective function. When it cannot find a better adjacent basic solution, it means that it has found the optimal solution. The criteria used by the simplex method are choosing the most efficient for the entering basic variable and the first basic variable that reaches zero for the leaving basic variable.

When all of the constraints of the model have a smaller than or equal to sign (≤), the first feasible basic solution is found from the slack variables. If this is not the case, we can apply the **artificial variable** technique to obtain an initial solution and later use the **two-phase method** to find the optimal solution.

In many models the variables are upper and lower bounded. By applying the **simplex techniques with bounded variables** we avoid having to increase the number of functional constraints of the problem. In the case of lower bounds, it is only required to make a change in the variable, while for variables with upper bounds it is also necessary modify the selection of the variable that leaves the basis at each iteration. The increase in **efficiency** is so considerable that the optimization software recommends identifying the bound constraints as this special type for as many variables as possible.

We have also commented on the most efficient way of implementing the simplex, the **revised simplex**, which is nothing other than a version of the simplex in which only the necessary data are calculated and stored at each iteration. In addition, this method allows the reduction of rounding errors. Lastly, the basic concepts of the latest developments regarding linear programming solving techniques have been described, in particular the **interior-point algorithm.** This starts from a point in the interior of the feasible region and it shifts in the direction in which the objective function is improved the quickest way, while maintaining feasibility. This process converges to the optimal solution. It is predicted that it will be used in combination with the simplex for large models with many thousands of functional constraints. Currently, the optimization software LINGO incorporates it (*Barrier Solver*).

3.7. SELECTED REFERENCES

1. Daellenbach, H.G.; George, J.A. and McNickle, D.C. (1987): *Introducción a las técnicas de Investigación de Operaciones.* CECSA.

2. Hillier, F.S. and Lieberman G.J. (2010): Introduction to Operations Research. Ninth Edition. McGraw-Hill.

3. Hillier, F.S and Hillier, M.S. (2011): Introduction to Management Science. A Modelling and Case Studies Approach with Spreadsheets. Fourth Edition. McGraw-Hill.

4. LINDO Systems (2011): *LINGO. The modeling language and optimizer*.

5. Winston, W. L. and S.C. Albright (2012): Practical Management Science. Fourth Edition. South-Western. USA.

3.8. CASE STUDIES

CASE STUDY 1

Solve the production planning model of section 3.4.1 without considering the constraints of minimum demand, and applying the simplex algorithm. Then answer the following questions from the final simplex tableau.

1. Indicate the optimal solution.
2. Are all resources completely used? If there are idle resources, indicate which ones and in what quantity.
3. The company is considering redistributing the resources between different departments. Analyze the convenience of the following alternatives and if you consider that there is a better one, indicate it. Explain your reasoning.
   a) Transfer 25 h/week from department 4 and 5 h/week from department 5 to department 3.
   b) Transfer 25 h/week from department 4 and 5 h/week from department 5 to department 1.
   c) Reduce the hours in departments 2 and 5.
4. Indicate the reduced costs of the variables and dual price of the constraints, and analyze the relationship of these with the slack variables.

CASE STUDY 2

Given the following linear programming model

$$\text{Max } Z = 24\,X_1 + 20\,X_2$$

$$0.5\,X_1 + X_2 \leq 12$$

$$1.5\,X_1 + X_2 \leq 24$$

$$0 \leq X_1 \leq 15$$

$$0 \leq X_2 \leq 10$$

1. Obtain the optimal solution by means of the graphic method.
2. Obtain the optimal solution by means of the simplex algorithm.
3. Find the optimal solution by applying the upper bound technique.

4. Identify each simplex tableau obtained in sections 2 and 3 with the corresponding corner-point in the graph carried out in the first section. Analyze the analogies and differences between the upper bound technique and the simplex algorithm.

5. Why is the upper bound technique more efficient?

CASE STUDY 3

Production planning models are among the best known applications of linear programming. However, as we know, what characterizes a linear programming model is not the context, but its mathematical structure. Although linear programming models are quite frequent in those problems in which the benefit of the company is maximized or the costs are minimized, the possibilities of linear programming are much wider and now we will see an example that illustrates this fact. This case study also serves as an example that allows you to graphically and algebraically solve a linear model where we *minimize* the objective function (in this chapter we explain the simplex algorithm for a maximization problem) and it contains the three types of constraints that can appear in a linear programming model ($\leq$, $\geq$, $=$).

The problem: design of a radiation therapy (Hillier and Lieberman, 2010)

When cancer is diagnosed in a very advanced stage, the only alternative with any expectancy of success is to apply radiation therapy in combination with chemotherapy and surgery. Radiation therapy consists of the use of a machine that passes ionizing radiation through the patient's body and damages both cancerous and healthy tissues.

Normally, the beams are administered accurately from different angles in a two-dimensional plane. Due to the attenuation, each beam delivers more radiation on the tissue near the entry point than on the point closer to the exit. Scatter also causes the tissues outside the trajectory of the beam to receive radiation. Because tumor cells are almost always microscopically spread among the healthy cells, the radiation dosage throughout the tumor region must be large enough to kill the malignant cells that are a little more sensitive to this, but small enough to avoid killing the healthy cells.

Simultaneously, the radiation dosage delivered to the critical tissues (vital organs) should not exceed the established levels of tolerance in order to prevent complications that can be more serious than the illness itself. For the same reason, the dosage that a healthy anatomy should receive has to be minimized.

The need to achieve a balance between all of these factors is what makes the design of a radiation therapy difficult. The main objective of the design is to choose the combination of beams to be used and their intensity to generate the best dose distribution possible. The strength of the dose at any point of the body is measured in *kilorads*. Once the treatment has been designed, it is administered in many sessions over several weeks.

For every beam with a given intensity the analysis of resulting radiation absorption in different parts of the body requires a complicated process. In summary, based on a careful anatomical analysis, the energy distribution within a two-dimensional cross section can be graphed in an isodose map in which the curves represent the strength of the dose as a percentage at its entry point. Then, a fine mesh is placed on the isodose map. If the radiation absorbed in the squares that contain each type of tissue is added, the average dose absorbed by the tumor, the healthy anatomy and the critical tissues can be calculated. The radiation absorption is additive when more than one beam is administered.

After such an exhaustive analysis, the medical team estimates, in detail, the necessary data for the design of the treatment. The summary is presented in Table 3.15, whose first column shows the areas that must be considered and the next two columns give the fraction of the dose of each beam that is absorbed on average in the respective areas. For example, if the dose level at the entry point for beam 1 is 1 kilorad, then 0.4 kilorads will be absorbed in the whole healthy anatomy in the two-dimensional plane, an average of 0.3 kilorads in the near critical tissues, an average of 0.5 kilorads in the different parts of the tumor and 0.6 will be absorbed in the center of the tumor. The last column gives the constraints on average on the total dose of both beams in different parts of the body. Thus, the dose received by the critical tissues should not exceed 2.7 kilorads, the average on the complete tumor should be approximately 6 kilorads and it should be at least 6 kilorads in the center of the tumor. In particular, the absorption average of the dose for the healthy anatomy should be as small as possible.

Table 3.15 Data for the design of the radiation therapy

Area	**Fraction of the entry dose absorbed by area (average)**		**Restriction on the total average dose**
	Beam 1	**Beam 2**	
Healthy anatomy	**0.4**	**0.5**	**Minimize**
Critical tissue	**0.3**	**0.1**	**≤ 2.7**
Region of the tumor	**0.5**	**0.5**	**= 6**
Center of the tumor	**0.6**	**0.4**	**≥6**

1. Formulate a linear programming model to solve this problem.
2. Obtain the optimal solution by means of the graphical method.

3. Solve the model by means of the Two-phase method.
4. Graphically, what is the effect of adding artificial variables to the ≥ and = constraints? Identify each simplex tableau obtained with its corresponding corner-point in the graphical method.
5. Which is the optimal dose? How much radiation does each one of the types of tissue under consideration receive? What would be the dose and the radiation received in each type of tissue if we wanted to the radiation received by the critical tissues to be reduced to 2.5 kilorads? What happens if you want to reduce to 2 kilorads? Answer these questions from the optimal simplex tableau.

The Memorial Sloan-Kettering Cancer Center located in New York City received the first prize of the Franz Edelman Award by INFORMS Society in 2007. This award recognized the application of mixed integer programming (with some binary variables) to optimize brachytheraphy (a procedure that involves placing radioactive material inside your body, sometimes called internal radiation) in prostate cancer that can be applied to other types of cancer.

CASE STUDY 4

Solve the following linear programming model using the upper bound technique and identify the U_j, β and δ at each iteration, and explain in detail all of the steps of the algorithm. What computational consequences does this procedure have when compared to the consideration of the variable bounds as functional constraints?

$$\text{Max } 10 X_1 + 15 X_2 - 10 X_3 + 25 X_4$$
$$2 X_1 + 2 X_2 + X_3 + 2 X_4 \leq 5$$
$$X_1 + 2 X_2 - 3 X_3 + 4 X_4 \leq 5$$
$$X_j \leq 1 \text{ for any } j = 1, 2, 3, 4$$

CASE STUDY 5

Given the following linear programming model of production, where X_1 and X_2 represent two products in thousands of units and R_1 and R_2 are the productive resources. Solve the model using the upper bound simplex technique. Explain in detail the selection of the entering and leaving basic variables in each iteration.

$$\text{Max } 7X_1 + 10 X_2$$
$$\text{R1: } 4X_1 + 6 X_2 \leq 15$$
$$\text{R2: } 3X_1 + 2 X_2 \leq 10$$

Maximum demand: $X_1 \leq 2$
Minimum demand: $X_2 \geq 1$

CASE STUDY 6: SOLID WASTE RECYCLING (Hillier and Lieberman, 2010)

A recycling plant includes 4 types of solid waste material and treats them to amalgamate them into a product that can be brought to market. The treatment and the amalgamated treatment are two different processes. Three different types of product can be obtained depending on the resultant mixture of materials which are used (Table 3 16.).

Quality standards require that there are minimum and maximum amounts of materials allowed in each type of product, expressed as a percentage of total product weight. These specifications are listed in Table 3.16, together with the cost of amalgamating and the selling price.

The plant collects waste materials from reliable sources so that stable production rates can be maintained to treat them. Table 3.17 shows the available weekly quantities, and the treatment cost.

The plant is owned by an environmental organization that has made contributions of €30,000 per week to be used only to cover the cost of full treatment of solid waste. The guidelines of the environmental organization are that the plant distributes the funds among plant materials so as to collect and treat at least half of each type of material.

Table 3.16. Specifications and amalgamated cost and sale price of the products

TYPE OF PRODUCT	SPECIFICATIONS	AMALGAMATION COST €/KG.	SALE PRICE €/KG.
A	Material 1 ≤ 30 % Material 2 ≥ 40 % Material 3 ≤ 50% Material 4 = 20 %	3	8,5
B	Material 1 ≤ 50 % Material 2 ≥ 10 % Material 4 = 10 %	2,5	7
C	Material 1 ≤ 70 %	2	5,5

Table 3.17. Availability and cost of material treatment

MATERIALS	MATERIAL 1	MATERIAL 2	MATERIAL 3	MATERIAL 4
AVAILABILITY KG/WEEK	3,000	2,000	4,000	1,000
COST €/KG	3	6	4	5

The plant manager wants to determine how much of each product type to produce and what mix of materials to maximize weekly profit (total revenue less cost of sales of the amalgamated product). The 30,000 euros per week that come from donations received by the environmental organization should be used to treat the materials. Formulate and solve the model for this problem. Also, explain in detail the information that provides the reduced costs in the optimal solution.

CASE STUDY 7

Solve the production planning model from section 3.4.1 with the additional constraints of maximum demands for the three products indicated.

$$
\begin{aligned}
&Max\ 20\,A + 18\,B + 21\,C \\
&0.2\,A + 0.1\,B + 0.3\,C \le 160 \\
&0.5\,A + 0.07\,C \le 80 \\
&0.1\,A + 0.3\,B + 0.1\,C \le 80 \\
&0.02\,A + 0.02\,B + 0.02\,C \le 40 \\
&0.05\,A + 0.06\,B + 0.05\,C \le 40 \\
&0 \le A \le 90 \\
&0 \le B \le 150 \\
&0 \le C \le 400
\end{aligned}
$$

1. Analyze and explain the differences between the optimal solutions of the three versions of the production planning model:

   1.1. Without demand constraints, solved in section 1.1 of case study1.

   1.2. With minimum demand constraints from section 3.4.1 of this chapter.

   1.3. With the maximum demand constraints from this section.

2. What is the best/worst for the company to increase the minimum or maximum demand in case of product A by 10 units? Why? Explain the reasoning from the optimal solution.

   2.1. Explain the information provided by the reduced costs of products A, B and C in the three versions of the problem.

   2.2. Identify the bottlenecks of the company in the three situations defined in section 1.

   2.3. Explain the value of the benefit obtained by the company from the concepts of dual price and reduced cost.

CHAPTER 4

DUALITY AND SENSITIVITY ANALYSIS

One hypothesis in linear programming is the certainty with respect to model parameters. In practice, model parameters are usually estimates. Therefore, when we have found the optimal solution for a problem, we have to analyze the effects of modifying the coefficients on the problem solution. For example, prices change with time and we may have used average prices in the model. In such a case, it would be necessary to evaluate whether the optimal solution would change if the price changed within the estimated limits. Fortunately, in linear programming it is very easy to carry out this study from optimal simplex tableaux.

This chapter focuses on how to carry out and interpret a sensitivity analysis. However, before studying sensitivity analysis we will introduce the concept of duality and the dual algorithm, which have important applications in economics. Finally, we will study parametric linear programming, since on many occasions the chosen values of some coefficients are only managing decisions. In this case, it is convenient to analyze the response of the problem solution to these decisions.

4.1. THE DUAL PROBLEM AND PRIMAL-DUAL RELATIONSHIPS

The problem of determining the **opportunity cost** of **resources** is also a linear programming model, which is actually the dual problem of the original problem. Duality is not only an interesting theoretical relationship. Let's describe it, as it is the basis of the concept of opportunity cost, of the dual simplex algorithm and of sensitivity analysis.

Every linear programming problem has another linear programming problem associated with it, and between them there are very special relationships. Each is the dual of the other. We shall illustrate it using the problem of production planning presented in section 3.4.1 of chapter 3, without regarding the demand constraints, i.e. the lower bound constraints of the variables.

4.1.1. THE PRIMAL PROBLEM AND THE DUAL PROBLEM

- Each problem constraint is associated with one variable of the other problem and vice versa.
- The technical coefficients of each problem constraint are the same as the technical coefficients of the corresponding variable in the other problem.
- The constraint's right-hand sides are the objective function coefficients of the corresponding variables in the other problem and vice versa.
- If we minimize in one problem with $\geq$ constraints and nonnegative variables, then in the other problem we maximize with constraints $\leq$ and nonnegative variables.

The ***dual problem*** associated with the production planning problem is shown in Table 4.1. In this Table we see that the original problem is referred to as the ***primal problem*.** Note that the dual problem has 5 variables, one for each constraint of the primal problem. It also has 3 constraints, one for each variable of the original problem, however, the objective function of the primal is to maximize and the corresponding objective function of the dual is to minimize. Note that the signs of the constraints are different in both problems and that the technical coefficients a_{ij} of a constraint in one problem are the associated variable in the other.

Table 4.1. The primal problem and the dual problem

ORIGINAL PROBLEM Resource allocation *Primal Problem*	NEW PROBLEM Resource price allocation *Dual Problem*
$A, B, C \geq 0$ Max $20\,A + 18\,B + 21\,C$ $0.20\,A + 0.10\,B + 0.30\,C \leq 160$ $0.50\,A + 0.07\,C \leq 80$ $0.10\,A + 0.30\,B + 0.10\,C \leq 80$ $0.02\,A + 0.02\,B + 0.02\,C \leq 40$ $0.05\,A + 0.06\,B + 0.05\,C \leq 40$	$W_1, W_2, W_3, W_4, W_5 \geq 0$ Min $160\,W_1 + 80\,W_2 + 80\,W_3 + 40\,W_4 + 40\,W_5$ $0.20\,W_1 + 0.50\,W_2 + 0.10\,W_3 + 0.02\,W_4 + 0.05\,W_5 \geq 20$ $0.10\,W_1 + 0.30\,W_3 + 0.02\,W_4 + 0.06\,W_5 \geq 18$ $0.30\,W_1 + 0.07\,W_2 + 0.10\,W_3 + 0.02\,W_4 + 0.05\,W_5 \geq 21$

4.1.2. PRIMAL-DUAL RELATIONSHIPS

The general relationships between the structures of the primal and dual problems are summarized in Table 4.2. There are also relationships between the solutions of both problems, as indicated below.

- If both the **primal** and **dual** problems have **feasible solutions**, then both have finite **optimal solutions** and the optimal **Z** values are **equal**.

- **Complementary slackness theorem**: If one constraint of either of the two problems has slack in any optimal solution of that problem, then in the other problem the variable associated with that constraint is zero for any optimal solution. If one variable of either of the two problems is not zero, then in the other problem the associated constraint is strictly fulfilled. This theorem indicates that a resource which is not used completely has a dual price of zero and a **resource** with a **dual price different from zero is scarce**.

Table 4.2. Relationships between primal and dual problems

PRIMAL PROBLEM	DUAL PROBLEM
$Maximize\ \ Z\ =\sum_{j=1}^{n} c_j x_j$	$Minimize\ \ Z\ =\sum_{i=1}^{m} b_i w_i$
Coefficients of Objective Function	*RHS Coefficients*
Coefficients Row i	*Coefficients Colum j*
Constraint ≤ $\sum_{j=1}^{n} a_{1j} x_j \ \leq b_1$	*Nonnegative Variable* $w_1 \geq 0$
Constraint ≥ $\sum_{j=1}^{n} a_{2j} x_j \ \geq b_2$	*No positive Variable* $w_2 \leq 0$
Constraint = $\sum_{j=1}^{n} a_{3j} x_j \ = b_3$	*Free Variable* $-\infty \leq w_3 \leq \infty$
Free Variable $-\infty \leq x_1 \leq \infty$	*Constraint* = $\sum_{i=1}^{m} a_{i1} w_i \ = c_1$
No positive Variable $x_2 \leq 0$	*Constraint* ≤ $\sum_{i=1}^{m} a_{i2} w_i \ \leq c_2$
Nonnegative Variable $x_3 \geq 0$	*Constraint* ≥ $\sum_{i=1}^{m} a_{i3} w_i \ \geq c_3$

4.2. DUAL SIMPLEX ALGORITHM

The simplex algorithm starts from a basic feasible solution in which every $X_j \geq 0$. It approximates the optimal solution maintaining feasibility at each iteration. When all $(C_j-Z_j) \leq 0$ an optimal solution for maximizing is found ($(C_j-Z_j) \geq 0$ for minimizing).

Sometimes we have initial solutions that are infeasible in the primal and feasible in the dual. That is, we have some $X_j \leq 0$, but all $(C_j-Z_j) \leq 0$ for maximizing ($(C_j-Z_j) \geq 0$ for minimizing). The dual simplex algorithm leads to primal feasibility, maintaining dual feasibility ($(C_j-Z_j) \leq 0$).

It is important to emphasize the fact that the dual simplex algorithm is a method used to solve any kind of problem, regardless of whether it is primal or dual, although we have explained the algorithm using the dual problem of the production planning model. The only requirement is to have a infeasible primal and a feasible dual solution.

Let's consider the dual problem of the production planning model, which is the following:

$$\text{Min} \quad 160\, w_1 + 80\, w_2 + 80\, w_3 + 40\, w_4 + 40\, w_5$$

$$\begin{aligned}
0.20\, w_1 + 0.50\, w_2 + 0.10\, w_3 + 0.02\, w_4 + 0.05\, w_5 &\geq 20 \\
0.10\, w_1 + 0.30\, w_3 + 0.02\, w_4 + 0.06\, w_5 &\geq 18 \\
0.30\, w_1 + 0.07\, w_2 + 0.10\, w_3 + 0.02\, w_4 + 0.05\, w_5 &\geq 21
\end{aligned}$$

$$w_1, w_2, w_3, w_4, w_5 \geq 0$$

We denominate w_6, w_7 and w_8 to the slack variables of the three constraints. Table 4.3, corresponding to the first simplex tableau, shows that the slack variables do not form the unity matrix. By multiplying the three constraints by (-1) we obtain that matrix, but the initial solution is infeasible, because on setting to zero, $w_1=w_2=w_3=w_4=w_5=0$, the slack variables take negative values. We know that in this type of situation we can apply the artificial variable technique. However, in this case we are going to apply the **dual simplex algorithm**, which results from **applying the criteria of the primal algorithm to the dual problem.**

CRITERION OF THE SIMPLEX DUAL ALGORITHM 1: LEAVING BASIC VARIABLE

The leaving basic variable X_i is the basic variable with the most negative value

OPTIMALITY CRITERION

The solution associated with a basic variable is optimal if every $X_i \geq 0$

CRITERION OF DUAL SIMPLEX ALGORITHM 2: ENTERING BASIC VARIABLE

> ***The entering basic variable is the variable with the lowest quotient $(C_j\text{-}Z_j)/ \text{-}\alpha^-_j$ if we minimize and the highest quotient if we maximize.***
>
> ***Minimization: Min $(C_j\text{-}Z_j)/ \text{-}\alpha^-_j$***
>
> ***Maximization: Max $(C_j\text{-}Z_j)/ \text{-}\alpha^-_j$***

As opposed to the simplex algorithm, first we have to select the leaving basic variable (the most negative) and then the entering basic variable, i.e. the variable capable of increasing the value of the basic variable in the most efficient way. In the illustrating example, w_8 leaves and w_1 enters because w_1 is the variable that involves the lowest increase in the objective function per unit increase in w_8. When comparing Table 4.3 with the first simplex tableau of the primal problem (case study 1 in chapter 3), we can see that the column of the entering basic variable corresponds to the row of the leaving basic variable. In the same way, the row of the leaving basic variable in the first iteration of the primal problem solution now corresponds to the column of the entering basic variable. The pivot element is the same, with its sign changed. Table 4.4 and following are obtained applying the same basic change procedures as in the simplex algorithm, given by equations (17) in chapter 3.

Table 4.3. Dual algorithm: initial simplex tableau

BASIC V.	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	b_i
W_6	-0.20	-0.50	-0.10	-0.02	-0.05	1	0	0	-20
W_7	-0.10	0,00	-0.30	-0.02	-0.06	0	1	0	-18
W_8	-0.30	-0.07	-0.10	-0.02	-0.05	0	0	1	**-21**
C_j-Z_j	160	80	80	40	40	0	0	0	0
C_j-Z_j/-α^-_{ij}	160/0.30 = **533.33**	80/0.07 = 1142.8	80/0.1 = 800	40/0.02 = 2000	40/0.05 = 800				

Table 4.4. Dual algorithm: second simplex tableau

BASIC V.	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	b_i
W_6	0	-0.45	-0.034	-0.01	-0.02	1	0	-0.66	-6.14
W_7	0	0.02	-0.27	-0.01	-0.04	0	1	-0.33	**-11.07**
W_1	1	0.23	0.33	0.07	0.17	0	0	-3.33	69.93
C_j-Z_j	0	42.66	26.67	29.33	13.33	0	0	533.33	-11200
C_j-Z_j/- α^-_{ij}			26.67/0.27 = **98.77**	29.3/0.01 = 2933	13.3/0.04 = 333.25			533.3/0.3 = 1616.5	

Table 4.5. Dual algorithm: third simplex tableau

BASIC V.	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	b_i
W_6	0	-0.45	0	-0.01	-0.02	1	-0.12	-0.62	**-4.81**
W_3	0	-0.07	1	0.04	0.15	0	-3.70	1.22	40.96
W_1	1	0.25	0	0.06	0.12	0	1.22	-3.73	56.42
C_j-Z_j	0	44.63	0	28.34	9.38	0	98.77	500.73	-12293.38
C_j-Z_j/- α^-_{ij}		**99.17**		2834	469		823.08	807.63	

Table 4.6. Dual algorithm: optimal simplex tableau

BASIC V.	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	bi
W_2	0	1	0	0.02	0.04	-2.22	0.27	1.37	10.58
W_3	0	0	1	0.04	0.15	-0.15	-3.68	1.3	41.68
W_1	1	0	0	0.05	0.11	0.55	1.15	-3.94	53.77
C_j-Z_j	0	0	0	27.42	7.39	99.17	86.80	439.14	-12770.38

If the tables obtained are compared with the tables that result from applying the primal simplex algorithm to the original problem, it can be noted that the dual algorithm comes from applying the simplex criteria to the dual problem. However, we must take into account that the dual algorithm serves to solve either of the two problems, as long as the initial solution is infeasible primal and feasible dual. We know that we can also use the two-phase method for infeasible initial solutions, but in this case it is not necessary for it to be feasible dual.

If we analyze the tables of the optimal solution, both for the primal problem and the dual problem, we verify that the **opportunity costs** of the **resources** in the production planning problem are given by the **dual variables**. We can also see that only the completely consumed resources, i.e. the scarce resources have a non-zero opportunity cost in the optimal solution.

When analyzing the structure of both linear programming models, we can see that in the primal problem we determine the utilization of resources to know what and how much to produce in order to maximize the company's benefits. That is, we can approach the problem from the point of view of business production or returns. On the other hand, the dual problem considers the distribution or sharing of that income. Therefore, the dual objective function consists of minimizing the value of the resources used by the company. The constraints make the value of the resources employed in manufacturing one product unit exceed or equal the product unit margin. Does this mean that the evaluation of the resources obtained by the dual program forces the company's benefit to be zero or negative? No, not at all. The complementary slackness theorem allows us to know that if one of the constraints takes a value higher than the second member, then the associated product would not occur. Therefore, if it costs more for a company to manufacture a product than the benefits obtained from it, the company will not be interested in manufacturing that product.

Dual variables can be interpreted as a mechanism that allows us to assign the gross margins of the products in the different resources. Verify that the unitary benefit of A can be attributed as 54 % to department 1, 25.6 % to department 2 and 20.4 % to department 3. In nonlinear programming, a similar result is obtained with the economic interpretation of Kuhn-Tucker conditions.

Finally, we have to take into account that dual variables do not always present a simple interpretation. Their interpretation will depend on the problem under analysis.

4.3. SENSITIVITY ANALYSIS OF THE COEFFICIENTS OF THE OBJECTIVE FUNCTION

The sensitivity analysis of the coefficients of the objective function consists of analyzing the effects of the changes in parameters C_j on the optimal solution. In chapter 2 we studied this effect graphically with the problem of energy generation in a thermal power plant. Remember that changes in one of these coefficients causes changes in the isoproduction slope. If the modification is large enough, the optimal solution may be another corner point of the feasible region (Figure 2.4). Therefore, changes in the Cj parameters may affect the optimality of the present solution, yet not affect its feasibility. How will a change in a C_j parameter affect the optimal simplex tableau? Only (C_j-Z_j) will change.

Chapter 2 presents the optimal solution and sensitivity analysis for the energy production model. The results indicate the range of values within which a coefficient of the objective function can be changed without changing the basis. The analysis is made by changing one coefficient each time, and maintaining all other coefficients constant.

Let's see how these ranges are determined with the simplified example of energy production, i.e. using only the smoke and pulverizer constraints, as we did in chapter 3 to explain the simplex algorithm. Figure 4.1 represents the feasible region and the optimal solution of this problem. It is necessary to emphasize that the optimal solution is the same, but the feasible region is now the polygon OABC. This feasible region is different from that of the original problem because two constraints have been eliminated.

As we have already mentioned, when changing a C_j the only elements that are affected in the optimal simplex tableau are the values of the C_j-Z_j. In order to determine the interval of C_j we have to take into account whether the parameter corresponds to a basic or non-basic variable.

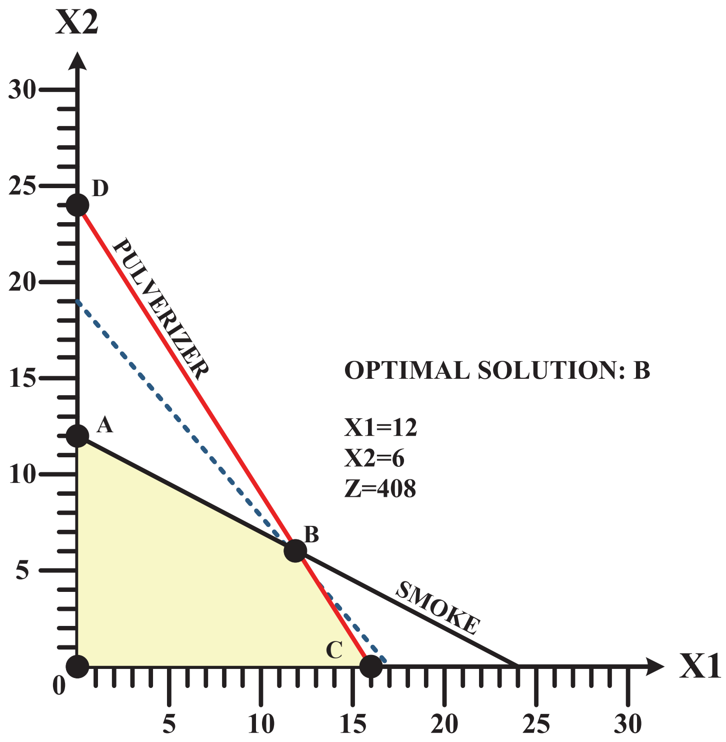


Figure 4.1. Feasible region and optimal solution for the simplified problem of energy production

4.3.1. MODIFICATION OF A C_j CORRESPONDING TO A NONBASIC VARIABLE

Let's see what happens if we modify C_3 in Table 3.3. No Z_j would change, since this variable is non basic. The only element affected will be its own C_j-Z_j, i.e. C_3-Z_3, due to the modification of C_3. Therefore, C_3 can be modified without changes in the optimal solution, as long as $C_3-Z_3 \leq 0$. As $Z_3=6$, then $C_3 \leq 6$. Thus, the range required is $-\infty \leq C_3 \leq 6$.

4.3.2. MODIFICATION OF A C_j CORRESPONDING TO A BASIC VARIABLE

For example, C_1. A modification to C_1 will affect all Z_j and, therefore, the values of the reduced values C_j-Z_j, except for the values of the basic variables, which are always zero. The interval of C_1 will determine the values for which $(C_j-Z_j) \leq 0$ for all nonbasic variables.

$$C_3-Z_3 = 0 - (3/2)\ 20 + C_1 \leq 0 \quad \Rightarrow \quad C_1 \leq 30$$

$$C_5-Z_5 = 0 + (1/2)\ 20 - C_1 \leq 0 \quad \Rightarrow \quad C_1 \geq 10$$

Graphically, when C_1=30, the objective function will run parallel to the pulverizer line and point B, point C and all midpoints will be optimal solutions (Figure 4.1). For C_1>30 only C will be optimal. Similarly, when C_1=10, the objective function will run parallel to the smoke constraint and point A, point B and all the points in the segment joining these two points will be optimal solutions. When C1 < 10, only point A will be an optimal solution.

In terms of the problem, C_1 represents the steam production with coal A in thousands of pounds/ton, therefore, as the combustion of coal A generates between 10,000 and 30,000 lb/ton, all the other parameters of the model being constant, the optimal solution will be to burn 12 ton/hour of coal A and 6 ton/hour of coal B. Obviously, the optimal value of the objective function will change in function of C_1. This interval for C_1 is very high; therefore it is unlikely that real steam generation lies outside the interval. Since measurement errors may occur, it becomes useful to know the range of values for the coefficients of the objective function in which the basis does not change. Determine the interval for coefficients C_2 and C_5.

4.3.3. SIMULTANEOUS MODIFICATIONS OF SEVERAL COEFFICIENTS

Let's now see simultaneous modifications of more than one coefficient of the objective function. The thermal plant engineers discovered an important error in the measurement of the steam generated, for both coal A and coal B. The valid coefficients are C_1=28 and C_2=18. Is the optimal solution of Table 3.3 still optimal? In this case, we cannot use the intervals for each coefficient individually, but we can apply the same procedure.

Since X_1 and X_2 are basic variables, changes in C_1 and C_2 can affect the (C_j-Z_j) of the nonbasic variables. Let's see if they are still non positive.

$$C_3-Z_3 = 0 - (3/2)\ 18 + 28 = -27 + 28 = 1$$

$$C_5-Z_5 = 0 + (1/2)\ 18 - 28 = -19$$

Therefore, the solution X_1 = 12 and X_2 = 6 is no longer optimal. By introducing the new values for C_j in Table 3.3 we get Table 4.7. From this table, we apply the simplex algorithm entering basic variable X_3 because it improves the objective function. X_2 leaves the base because it is the only variable that can be eliminated. The result is Table 4.8 which corresponds to the optimal solution for the new situation. This solution consists of burning only coal A, specifically 16 ton/hour, the steam generated is 448 thousand pounds, the pulverizer is made to work at its maximum capacity and the maximum level allowed for smoke emissions is not reached.

Table 4.7. Determination of the optimal solution after changing several C_j

BASIC VAR.	X_1	X_2	X_3	X_5	b_i
X_2	0	1	3/2	-1/2	6
X_1	1	0	-1	1	12
$C_j - Z_j$	0	0	1	-19	-444

Table 4.8. Optimal Solution

BASIC VAR.	X_1	X_2	X_3	X_5	b_i
X_3	0	2/3	1	-1/3	4
X_1	1	2/3	0	2/3	16
$C_j - Z_j$	0	-2/3	0	-18,66	-448

In practice it is very useful to apply the **100% rule** for **simultaneous changes** in the coefficients of the objective function. If changes are made simultaneously in several C_j, for each change, the percentage represented on the allowable change (increase or decrease) for the values of variables in the optimal solution not to change, is estimated. If the sum of the percentage of change does not exceed 100%, the original optimal solution will remain optimal and if the sum exceeds 100% there is no certainty that it will still be optimal.

When applying the 100% rule to the previous case where $C_1 = 28$ and $C_2 = 18$ we have the following percentages:

((28-24) / 6) *100 = 66 %

((20-18) / 4) * 100= 50 %

C_1 is increased by 66% without changing the values of the variables in the optimum value and C_2 decreases by 50%, as the sum exceeds 100% we cannot be sure that the values of the variables are kept at the optimum. In fact, we checked with the formulas of the reduced costs that the optimal solution changes.

4.4. SENSITIVITY ANALYSIS OF THE RIGHT-HAND SIDE OF THE CONSTRAINTS

It is also important to determine the sensitivity of the optimal solution to changes in the independent terms of the constraints, because on many occasions its values represent a managerial decision. Let's remember that in section 2.5.2 the advisability of installing a system to reduce smoke emissions by 25% was mentioned. This means that the smoke constraint in this case would be

$$(1) \qquad 0.5\,X_1 + X_2 \leq 15$$

We also saw graphically, that modifying the second members of the constraints means shifting their lines in parallel (Figure 2.5). In particular, increasing the capacity of smoke emissions in one unit moves the constraint up, so that the optimal solution changes from point A to point D in Figure 2.5. The objective function experiences a variation equal to the opportunity cost of the smoke constraint.

Figure 2.5 shows us that there is an interval for the values of b_i, in which we can change this parameter without affecting the basic variables and, therefore, the associated opportunity cost remains constant. Over this interval, the opportunity cost of the smoke constraint is 6 and the activity levels of the variables change linearly along the pulverizer line.

To illustrate how the ends of the interval for b_i are calculated, we will again use the simplified energy production problem using only the smoke and pulverizer constraints. Therefore, the resulting range need not necessarily coincide with the range determined for the original problem. Why?

The range of values that b_1 can take without modifying the basis is obtained from the optimal simplex tableau, shown in Table 3.3. The modification of a b_i does not affect the optimality, since it does not modify the C_j-Z_j, but it can cause non feasibility. From Table 3.1 the smoke constraint will be

$$0.5\,X_1 + X_2 + X_3 = b_1 \qquad (2)$$

In the problem b_1=12 and for this value the optimal solution is X_1=12, X_2=6 and X_3=X_5=0 with Z=408. As b_1 increases, the optimal solution changes, moving up along the pulverizer constraint (Figure 4.1). The intersection between the smoke and pulverizer constraint lines indicates the optimal solution. This causes changes in the activity levels of the variables in the optimal solution, but the basis does not change. The same would happen if the second member of the smoke constraint decreased. There is an interval for which the basic variables are the same and their activity levels change linearly along the pulverizer line.

Since the smoke emission limit is 12, *(2)* it can be written as

$$0.5\,X_1 + X_2 + X_3 = 12 \qquad (3)$$

In the optimal solution $X_3 = 0$, thus

$$0.5\,X_1 + X_2 = 12 + \Delta \qquad (4)$$

when b_1 increases or decreases. Equation *(4)* is equal to *(3)* with $X_3 =- \Delta$.

We can then **analyse the modification of the** right-hand **parameter** of a constraint, by analysing the change in the corresponding **slack variable**. If the equation does not present a positive slack variable, the **artificial variable** has the same effect.

From Table 3.3 the equations of the optimal tableau are:

$$X_2 + 3/2\,X_3 - 1/2\,X_5 = 6 \qquad (5)$$

$$X_1 - X_3 + X_5 = 12$$

$$- Z - 6\,X_3 - 14\,X_5 = - 408$$

setting $X_3 = - \Delta$, with $X_5 = 0$ the new values of the basic variables are then

$$X_2 - 3/2\,\Delta = 6 \Rightarrow X_2 = 6 + 3/2\,\Delta \qquad (6)$$

$$X_1 = 12 - \Delta$$

Since a modification of a b_i does not modify the C_j-Z_j, the current solution will be the optimal solution if feasible, i.e. all $X_j \geq 0$. Therefore, if we consider Δ^- as the maximum decrease and Δ^+ as the maximum increase, from *(6)* we get that

$$\Delta^- = 4 \qquad (7)$$

$$\Delta^+ = 12$$

To summarize, if b_1 lies between 8 and 24 the opportunity cost of the smoke constraint will be 6.

In a general sense, the intervals for the second members of the constraints are calculated in the following way. If b_i is the i-th element of the column for the optimal simplex tableau solution and α_i is the i-th element of the column corresponding to the **slack or artificial variable for constraint k**, then

$$\Delta^{+i} \text{ in } b_k = \text{Min } (- bi/\alpha_i) \text{ for any } \alpha_i < 0 \text{ and } +\infty \text{ if every } \alpha_i \geq 0 \qquad (8)$$

$$\Delta^- \text{ in } b_k = \text{Min } (bi/\alpha_i) \text{ for any } \alpha_i > 0 \text{ and } +\infty \text{ if every } \alpha_i \leq 0$$

Now suppose that the regulations for polluting emissions are tightening and b_1=7 is established. What would the optimal solution be now? Upgrading column b_i of the optimal simplex tableau we get table 4.9 with X_3 = 5.

Table 4.9 Resulting simplex tableau after decreasing b_1 to 7 units

BASIC VAR.	X_1	X_2	X_3	X_5	b_i
X_2	0	1	3/2	-1/2	-1,5
X_1	1	0	-1	1	17
C_j - Z_j	0	0	-6	-14	-378

Applying the dual algorithm to Table 4.9, X_2 leaves and X_5 enters. The resulting tableau is table 4.10

Table 4.10 Optimal simplex tableau

BASIC VAR.	X_1	X_2	X_3	X_5	bi
X_5	0	-2	-3	1	3
X_1	1	2	2	0	14
Cj - Zj	0	-28	-48	0	-336

Note that by changing the basic variable the opportunity cost has changed as well, a fact to take into consideration when analyzing whether it is interesting to change b_i when this involves changes in the basic variable. In this case, the optimal opportunity cost may not be valid for all the increases or decreases studied.

We can also apply the **100% rule** for **simultaneous changes** in the second members of the constraints. If changes are made simultaneously in several b_i, for each case the represented changes are calculated on the allowed range (increase or decrease) so that it remains feasible. The change is calculated as if only one b_i would vary at a time. If the sum of all percentages of changes does not exceed 100%, opportunity costs will remain the same and if the sum exceeds 100% there is no certainty that this is the case

4.5. PARAMETRIC LINEAR PROGRAMMING

Parametric programming consists of obtaining the sequence of optimal solutions when we change the parameters of a model over a wide range. Continuing with the simplified problem of energy production, considering only the smoke and pulverizer constraints, suppose that we want to change C_1 from zero to infinity.

We have seen in section 4.3.2 how, when we vary C_1, the optimal solution to the problem will be point A, B or C in Figure 4.1. Table 4.11 shows the values of the variables in each case. When C_1 is less than 10, it is of interest to use only the second type of coal and when it is greater than 30, only the first type.

Table 4.11. Results of the parametric programming of C1.

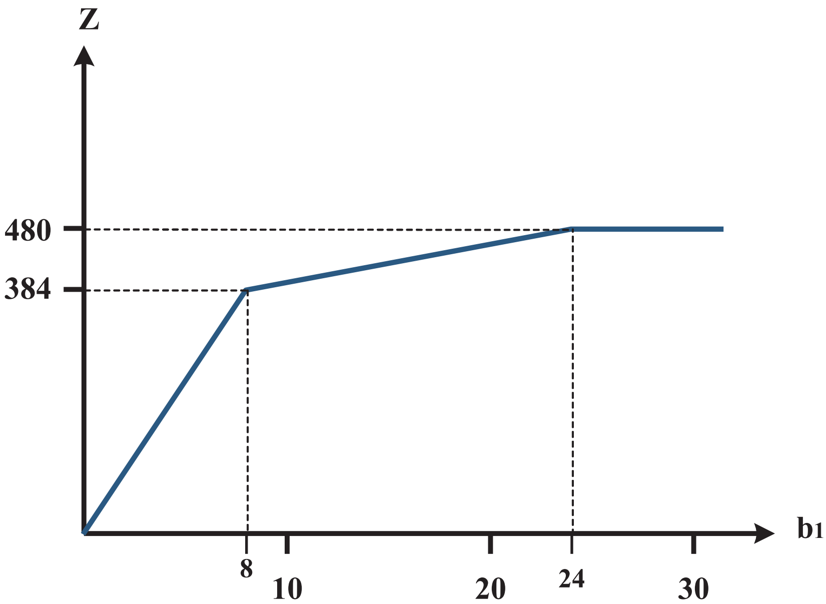
VARIABLE	$0 \leq C_1 \leq 10$	$10 \leq C_1 \leq 30$	$30 \leq C_1 \leq \infty$
X_1	0	12	16
X_2	12	6	0

In the parametric programming of independent terms of constraints the right-hand parameters are made to change over a wide range. The sequence of optimal solutions can be obtained by applying the dual simplex algorithm.

If in Figure 4.1 we changed b_1 from zero to infinity, the smoke constraint line would move parallel from the origin upwards. The optimal solution would hen change in a continuous manner first along segment OC, then along segment CD. In the section OC the basic variables are X_1 and X_5, while in the CD segment, X_1 and X_2 are basic. If we could increase the amount of smoke to more than 24 kg/hour, this constraint would not have an opportunity cost, because of the left over smoke emission capacity. The spraying machine has become a scarce resource (see Table 4.12).

Table 4.12. Results for the parametric programming of b_1

b_1	OPPORTUNITY COST	BASIC VARIABLES
0 - 8	48	X_1, X_5
8 - 24	6	X_1, X_2
$\geq$ 24	0	X_2, X_3



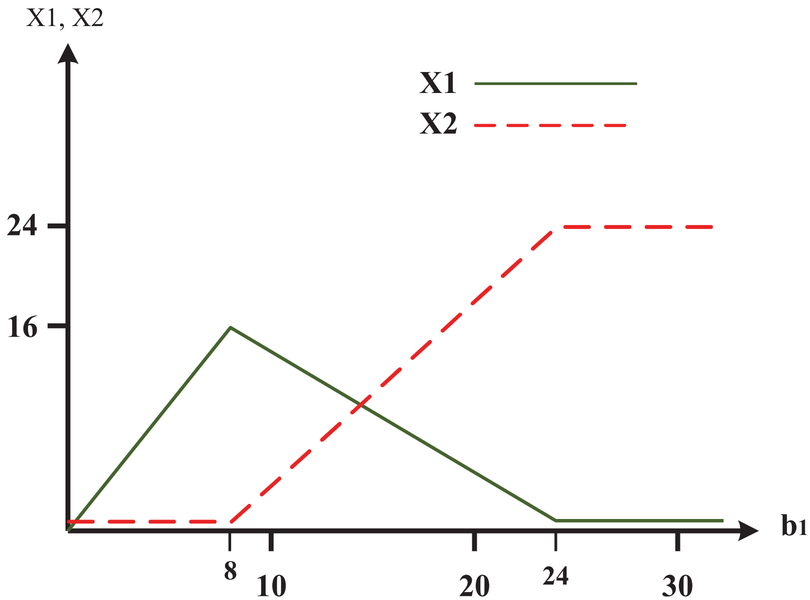


Figure 4.2. Parametric Programming of b_1: change in the objective function and activity levels of the variables

The results of the parameterization are shown in Figure 4.2. Both graphs display the three segments for the independent term of the smoke constraint. The only element which remains **constant** in each of them is the **opportunity cost**, as the objective function changes linearly, as does the activity level of the variables. Observe carefully the evolution of the objective function. This linear form by segments and concave is characteristic of parametric programming and highlights the **law of decreasing marginal profits of the resources.**

4.6. SUMMARY

We have seen that **duality** is the theoretical basis of the **opportunity cost** and of other important techniques in Operations Research, such as the **dual simplex algorithm** and **sensitivity analysis**. It allows us to evaluate which **modifications the parameters of the model can experience without affecting the optimal solution**.

The optimal simplex tableau allows us to calculate, with little computational cost, the range of values of the coefficients of the objective function that can be changed individually without changing the optimal activity levels of the variables. We have only to take into account that the C_j-Z_j must remain negative (maximization). Similarly, the ranges for the second members of the constraints are determined subject to the condition that the variables be nonnegative. When the changes cause non feasibility, the dual algorithm allows us to determine the optimal solution in an efficient manner.

Finally, we have seen that parametric analysis is just a generalization of sensitivity analysis that determines how the optimal solution evolves when the parameters of the model vary over a wide range. The dual simplex algorithm is a basic tool for this kind of analysis.

4.7. SELECTED REFERENCES

1. Daellenbach, H.G.; George,J.A. and McNickle,D.C. (1987): *Introducción a las técnicas de Investigación de Operaciones.* CECSA.
2. Hillier, F.S. and Lieberman G.J. (2010): Introduction to Operations Research. Ninth Edition. McGraw-Hill.
3. Hillier, F.S and Hillier, M.S. (2011): Introduction to Management Science. A Modelling and Case Studies Approach with Spreadsheets. Fourth Edition. McGraw-Hill.
4. Winston, W. L. and S.C. Albright (2012): Practical Management Science. Fourth Edition. South-Western. USA.

4.8. CASE STUDIES

CASE STUDY 1: DUAL PROGRAM AND DUAL ALGORITHM

1. Develop the dual program associated with the following primal program.

$$Max\ 3X_1 + 4X_2 + X_3$$

$$X_1 + 3X_2 + 2X_3 \geq 10$$
$$6X_1 + 2X_2 + X_3 \leq 30$$
$$X_1 + X_2 + X_3 = 5$$

$$X_1, X_2, X_3 \geq 0$$

2. Solve the primal problem of the previous section through the dual simplex algorithm.

3. Obtain the optimal solution of the dual program from the results obtained in point 2.

CASE STUDY 2: SENSIBILITY ANALYSIS

1. Determine the variation interval for coefficients C_2 and C_5 in the energy generation problem presented in chapter 2.

2. In the production planning problem solved in case study 1 in chapter 3 the benefits per unit of the three products have been revised. The new values are $C_A = 40$, $C_B = 30$ and $C_C = 12$. Is the current solution still optimal? If not, determine the new optimal solution from the optimal simplex tableau.

3. Determine the ranges over which b_1 and b_3 can change individually in the energy generation problem presented in chapter 2. Compare the result obtained for b_1 with the result obtained in section 4.4 and explain why they are different.

4. Determine the ranges over which the b_i of the following linear program can change individually (chapter 3-case study 2):

$$Min\ Z = 0.4X_1 + 0.5X_2$$

$$X_1, X_2 \geq 0$$

$$0.3X_1 + 0.1X_2 \leq 2.7$$
$$0.5X_1 + 0.5X_2 = 6$$
$$0.6X_1 + 0.4X_2 \geq 6$$

CASE STUDY 3: SENSIBILITY ANALYSIS

From the corresponding simplex table to the optimal solution of the model results of the case study 5 of Chapter 3, answer the following questions.

1. Indicate the optimal solution, the opportunity cost of resources and the variables reduced cost and their meaning.

2. By how much can the unitary profit of X2 vary without changing the optimal solution? Does the total profit vary? What does the sensitivity analysis of the objective function coefficients tell us?

3. By how much can the second member of the R1 constraint vary without changing the base? What changes and what remains constant while varying b_1? Why is the sensitivity analysis of the secondary members of the constraints important when making decisions on production problems?

CASE STUDY 4: FEED MANUFACTURING AND DISTRIBUTION

A multinational feed manufacturing and distribution company has 9 factories in the country. The factory located in Valencia produces183 formulas for 8 different species of animals. Every day, 43 trucks leave the factory to distribute 783,000 Kg. of feed. There are two types of feed: feed flour and feed granule. Feed is also sold in two different ways: in bulk or in sacks. The distribution of the different types of feed is shown in Table 4.13.

Of the quantity produced, two of the most important formulas represent a third of the total and the first six imply 55% of the total production. On the factory's premises an average of 30 T of feed in sacks and all of the feed in bulk is sold directly to clients. The factory has 647 clients and it receives an average of 80 orders per day, of which 6 are filled directly on site and 74 are sent by truck.

At the moment, the nearest farm to which feed is distributed is located 15 Km from the factory and the most distant is 340 Km. 35% of the trucks delivering in bulk serve only one client, while this percentage is 7% for the trucks that deliver feed in sacks. The average number of deliveries for trucks which distribute to more than one client is 2-3. For the trucks that distribute sacks, the average is 4-5.

The average delivery time of orders is 1.25 days, measured as the difference between the date on which the order is received and the real date of delivery. The person in charge of logistics takes 6 hours each day to prepare the routes for the following day, including checking stocks, availability of trucks, etc.

Table 4.13. Type of products and quantities sent from the factory

TYPE	NUMBER OF PRODUCTS IN the CATALOG	AMOUNT SENT DAILY (Kg)	NUMBER OF DAILY TRUCKS
Bulk Flour feed	1	2,000	
Bulk Granule feed	98	600,000	32
Flour in sacks	6	1,000	
Granule in sacks	78	180,000	11

Currently, production is scheduled according to an established production plan, taking pending orders into account. 60% of the daily production is immediately loaded onto the trucks.

There are three working shifts in the factory: from 6-14, from 14-22 and from 22-6. Production capacity is 37,000 Kg/hour and packaging 375 sacks/hour. The factory has 12 of 36,000Kg capacity containers and 17 of 17,000Kg capacity containers for bulk storage. The storage capacity for sacks is 4 cells of 32,000 Kg capacity. The real amount of stock available is checked twice a day.

Tables 4.14 and 4.15 show the information related to the technical characteristics of the raw materials, cost, as well as the nutrition needs of each type of feed according to the species and the age of the animal.

Outline a linear programming model that allows to determine the formulation and cost of 100 T of feed for lambs P1 and 100 T of feed for lambs P3.

1. Do you consider that the optimal formulation that you can obtain with optimization software could be composed of 20 raw materials? Explain your answer.
2. Solve the previous problem and indicate the optimal composition and the cost of feeds P1 and P3.
3. Barley is not part of any of the two previous formulations. Do you believe that it would be part of one of them if its price were 25 m.u./Kg? And if its price were 22.5? What would happen if the prices of barley and wheat decreased by 1 m.u./Kg? First, answer without solving the model again and then check if your answer is correct.
4. Analyze the sensitivity of the optimal to the prices of the remaining raw materials and the utility of this information to the company.

5. If the price of corn were increased by 0.2 m.u./Kg. and that of soy decreased by 0.5 m.u./Kg., would the optimal formulation and the cost of feed P1 change? What about if soy were decreased by 1 m.u./Kg? Why? Check what changes and how different it is from the initial solution.

6. What would be the effect if feed P1 had a minimum level of gross protein of 18.5 and of 1.10 of calcium? Answer applying the 100% rule.

7. Calculate the optimal formulation of minimum cost that takes into consideration that feed P3 cannot have more than 40% corn, 10% gluten and 5% sunflower.

8. This type of company usually has to face limited stocks of raw materials. Determine the cost and the formulations of P1 and P3 from the previous section in the following cases:

   8.1. There is only a stock of corn of 50 T.

   8.2. The price of the 50 T of corn in stock is 26.2 m.u./Kg and the company can buy in the market as much corn as they wish at 27 m.u./Kg.

   8.3. The company has 50 tons of corn in stock that it has to consume. The company can also acquire all the corn they want at 24 m.u. /Kg.

Evaluate in each section the consequences there could be for the company to solve the problem by means of simple formulation instead of using a multiformulation model, supposing that half of the production were P1 and the other half P3.

9. Obtain the formulation and the cost of P1 and P3 in such a way that if tapioca, gluten and molasses are part of any of the feed they have a minimum level of 10%. Use the semi-variable option of LINGO. How would you solve it with Solver Excel spreadsheet? Integer variables are needed to incorporate such a situation to a model.

Table 4.14. Nutritional limits of feed for lambs

CHARACTERISTICS	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
U.F.V/Kg Min	1.07	1.02	1.00	1.02	1.02	1.04	0.78	0.98
GROSS PROT. % Min	18.0	17.6	17.2	17.6	17.6	18.0	14.0	16.5
GROSS PROT. % Max	19.0	18.6	18.2	18.6	18.6	19.0	15.0	17.5
GROSS FIBR. % Min	-	-	-	-	-	-	14.0	-
GROSS FIBR. % Max	10	4.40	4.30	4.40	4.4	4.5	15.0	-
FAT MAT % Min	-	-	-	-	-	-	-	-
FAT MAT % Max	5.30	4.40	4.30	4.40	4.4	4.50	-	-
STARCH % Min	-	35.0	35.0	35.0	35.0	35.0	-	30.0
CALCIUM % Min	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
CALCIUM % Max	1.30	1.10	1.10	1.10	1.10	1.10	1.10	1.10
PHOSPHORUS T.% Min	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
PHOSPHORUS T.% Max	0.44	0.35	0.35	0.35	0.35	0.35	0.35	0.35
M.N.D. % Min	-	-	-	-	-	-	-	-
P.D.I.E. % Min	13.0	12.1	11.9	12.1	12.1	12.3	-	11.2
P.D.I.N. % Min	13.0	12.1	11.9	12.1	12.1	12.3	-	11.2

CHARACTERISTICS	CORN	WHEAT	BARLEY	SORGHUM	BRAN	MANIOC	GLUTEN FEED	STRAW	MOLASSES	ALFALFA	PEAS
COST (m.u.)	26.2	25.75	23.2	35	18	18.77	19.1	12.8	12.25	19.45	60
U.F.	1.125	1.047	0.9980	1	0.736	0.97	0.88	0.37	0.7	0.54	1
GROSS PROT.	8.5	11.05	11	9.4	15.5	2	9.5	4	4	17	20.5
GROSS FIBER.	2.1	2.4	4.7	2.2	8.2	3	10	37	-	25	5.7
FAT. MAT.	3.7	1.6	1.7	3	4	0.4	1.8	2.4	-	2.5	1.5
STARCH	64	60	50	64	18	65	25	-	-	3.2	41.9
CALCIUM	0.02	0.05	0.11	0.03	0.13	0.1	0.3	0.5	0.7	1.55	0.08
PHOSPHORUS	0.27	0.33	0.3	0.26	0.92	0.8	0.7	0.1	0.07	0.25	0.4
M.N.D.	6.05	9.4	7.7	6.2	11.9	1.04	16.38	-	2	11.7	18
P.D.I.E.	10.25	8.8	9.38	10.9	8.89	6.7	10	4	4.4	11.5	8.61
P.D.I.N.	6.2	7.4	7.36	7.3	10.1	1.8	13.1	2.2	2.2	11.5	12.8

Table 4.15. Characteristics of the raw materials: cost and nutrients

CHARACTERISTICS	SOYA BEAN	SOYA	SUNFLOWER	RAPE	BUTTER	TALLOW	OLEINS	SERUM	UREA	PHOSPHATE	CARBONATE	SEPIOLITE
COST (m.u.)	40	37	16.6	23.85	72	61.8	49.5	57	35.5	40.4	3	6.6
U.F.	1.24	0.995	0.71	0.75	3	3	-	1.15	-	-	-	-
CRUDE PROT.	37	44.4	36	38	-	-	-	11	285	-	-	-
CRUDE FIBER	5.5	6.2	17.8	12.3	-	-	-	-	-	-	-	-
FAT. MAT	20	1.25	2.4	2	99.4	99	98	0.7	-	-	-	-
STARCH	3	4	3.5	5	-	-	-	-	-	-	-	-
CALCIUM	0.2	0.25	0.3	0.75	-	-	-	1.65	-	25.5	38.6	-
PHOSPHORUS	0.5	0.62	1	1.15	-	-	-	0.85	-	17	-	-
M.N.D.	30.9	40.85	31.18	31.68	-	-	-	9.9	200	-	-	-
P.D.I.E.	21	24.2	13.85	14.67	50	50	-	8.49	-	-	-	-
P.D.I.N.	27	31.7	23.4	24.39	-	-	-	6.73	159.6	-	-	-

Table 4.15. Characteristics of the raw materials: cost and nutrients (continuation)

CHAPTER 5

INTEGER PROGRAMMING

In linear programming all variables are continuous and if they represent quantities such as time, Kg. of raw material or Euros, it is an optimal representation. However, when the variables which need to be rounded to the next integer refer to discrete elements such as machines or people, we can move a long way from the optimal, except in those cases in which the variables present high values this procedure is acceptable. In short, the *divisibility* hypothesis limits the field of application of linear programming, since in many real-life problems the variables only make sense if their value is integer.

After noting the need to turn to integer programming with a simple example, in this chapter we will learn to formulate integer programming models for solving problems with yes-or-no decisions, in which there are decisions such as to make an investment or not, locating an industry/warehouse/service at a particular site or not or when the cost function is nonlinear, among many other cases. We will also address one of the resolution techniques currently most used by optimization software: the Branch-and Bound algorithms that use the simplex method discussed in Chapter 3. These techniques are now being integrated with a technique from computer science known as *constraint programming* that promises to expand the capacity to formulate and solve integer programming problems.

5.1. INTRODUCTION

The mathematical model of integer programming is similar to that of linear programming with the additional constraints that some or all variables must be integer. It is called ***pure integer programming*** if all variables are integer; ***mixed integer programming*** if only some variables are integer and ***binary integer programming*** when all variables of the model are binary or (0-1) variables.

The additional condition that the variables have to be integer makes solving the problem considerably more difficult. Why? After all, it is nothing more than linear programming with fewer solutions to consider. In pure integer programming with a limited set of possible solutions, like that in Figure 5.1, we have guaranteed the existence of a finite number of solutions. However, this fact does not mean that the problem can be solved, since finite numbers can be astronomically large. Thus, in a binary programming problem with n variables there are 2^n solutions (some of them can be discarded as they violate the functional constraints). Every time that n increases by one, the number of solutions is doubled. This pattern is called exponential growth of problem difficulty. With n=10 there are more than one thousand solutions (1024), with n=20 more than one million, with n=30 more than one thousand million and so forth. Therefore, even the most efficient computers are unable to carry out an exhaustive enumeration, verifying first that the solution is feasible and then calculating the value of the objective function.

If we look at Figure 5.1, which represents the set of feasible solutions for a linear programming model with two integer variables, we see that the feasible region is not a convex set. We know that this property is basic in linear programming theory and the fact that it is not present in integer programming hinders the solving method. Nevertheless, as we will see in this chapter, many techniques widely use the simplex algorithm or dual simplex algorithm.

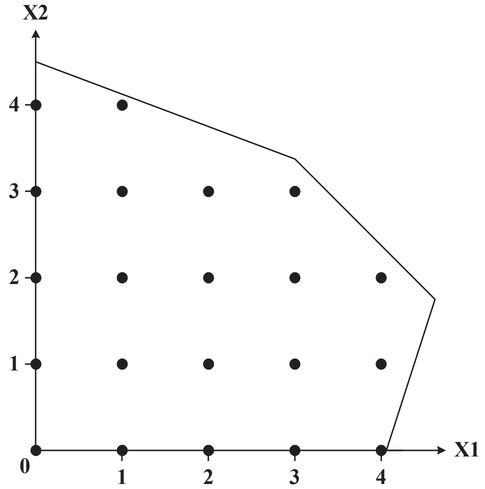


Figure 5.1. Feasible set in integer programming

5.2. A SIMPLE PROBLEM TO DISTRUST OF ROUNDINGS

A department has received €250,000 to acquire new equipment and it is impossible to increase this amount with funds from other sources. Several studies point out that only two types of machines are appropriate and that any number or combination of those machines would be acceptable. Some tests have been carried out to evaluate the load capacity in units of "average number of jobs" per hour for both types of equipment.

Table 5.1. Cost and capacity of equipment

Equipment	**Cost** In thousands €	**Capacity** Number of jobs/hour
1	140	28
2	60	11

The objective of the department is to maximize the potential working capacity. It is evident that the equipment can only be acquired in integer units.

If X_1 is the number of equipment type 1, and X_2 the number of type 2, the capacity of the number of jobs per hour is 28 X_1 + 11 X_2. Therefore, the objective function is

$$Max\ Z = 28\ X_1 + 11\ X_2$$

The constraints will refer to the available resources, nonnegativity conditions and the variables have to be integer.

Available resources:	$140\ X_1 + 60\ X_2 \leq 250$
Nonnegativity conditions:	$X_1, X_2 \geq 0$
Integer conditions:	X_1, X_2 *integers*

This problem has to be solved by integer programming because of the small values that the variables will take in the optimal solution. In Figure 5.2 we can see that the continuous optimal solution is $X_1 = 25/14$, $X_2 = 0$ and the value of the objective function is 50. If we applied the naive practice of rounding up to the nearest integer, we would obtain $X_1 = 2$, $X_2 = 0$ and $Z = 56$, that is, a solution that is infeasible. By testing the next feasible integer solution $X_1 = 1$, $X_2 = 0$ and $Z = 28$ are obtained. However, this solution is much worse than the **optimal integer solution** which is:

$$\boldsymbol{X_1 = 0}$$
$$\boldsymbol{X_2 = 4}$$
$$\boldsymbol{Z = 44}$$

Note that this solution has a jobs per hour capacity of 88% of the value obtained in the continuous solution, while the rounding to the next feasible integer result only gave us a working potential of 56%.

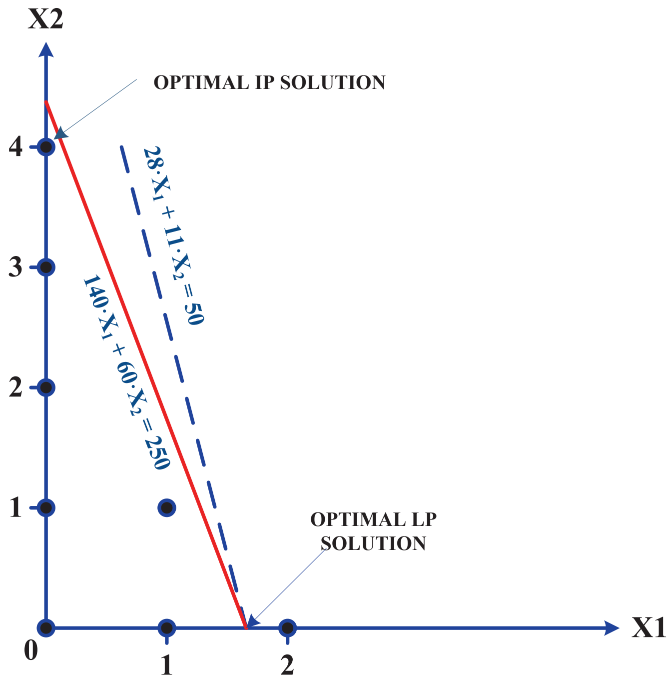


Figure 5.2. Feasible region and optimal solution for model of equipment acquisition

5.3. SOME APPLICATIONS OF INTEGER PROGRAMMING

Integer programming is used in all those problems that can be represented by means of a model that complies with all linear programming hypotheses, except for that of divisibility. We have already said that rounding to integers can be acceptable when the variables take high values, but this will not usually be the case in problems like the one shown in section 5.2 where the values of the variables are small.

There is another area of application, which is more important, that comprises all those problems in which yes-or-no decisions are presented. For example, should a specific project be started? Should an investment be made? Should an industry be located in a particular site? We will represent these situations by means of binary variables, that is, (0-1) variables. In such cases these variables are decision variables. However, binary variables are also used in integer programming as "auxiliary" variables whose objective is to formulate difficult problems so that they can be easily handled. Thus, we can present logical propositions as linear constraints or nonlinear functions can be handled as if they were linear.

5.3.1. CAPITAL BUDGETING DECISIONS

An automobile manufacturing company wants to calculate the optimal distribution of its capital budget for the next years. The future investments that it is considering are the following:

a) Renovating its current factory in order to increase its production capacity by 10,000 units per year.

b) Constructing a new factory, with a production capacity of 15,000 units per year.

c) Putting into practice the results that have been obtained from a survey on workers' motivation at work, which will increase the production by 5,000 units per year.

d) Purchasing more productive and technologically more advanced equipment, with which an increase of production of 2,000 units per year will be achieved.

Investments a and b are mutually exclusive, and investment d is conditional on the realization of investment a. The company seeks to increase its installed productive capacity, but considers that, due to the maximum sales forecasts, it should not exceed an increase of 16,000 units per year.

The Net Present Value (NPV) of the investments, as well as the expenditure and the financial availability are those shown in table 5.2.

Table 5.2. Investment data

Investment	NPV	Cash-flow	
		Year 1	Year 2
I1	50	60	70
I2	80	100	70
I3	40	30	40
I4	30	20	50

The problem consists of finding out which investments should be carried out in order to optimize their net present value, knowing that the financial availabilities for the first and second period are 150 and 120 m.u. respectively.

The integer programming model to solve this problem is the following:

Variables

We define four binary variables $\boldsymbol{X_j = (0, 1)}$ for $\boldsymbol{j = 1, 2, 3, 4}$ that will take the value 1 when the investment j is carried out and 0 otherwise.

Objective function

The objective is to maximize the Net Present Value of all investments.

$$Max\ 50\,X_1 + 80\,X_2 + 40\,X_3 + 30\,X_4$$

Constraints

The expenditure during the first and second year cannot exceed the availability:

$$60\,X_1 + 100\,X_2 + 30\,X_3 + 20\,X_4 \leq 150$$

$$70\,X_1 + 70\,X_2 + 40\,X_3 + 50\,X_4 \leq 120$$

Investments *a* and *b* are mutually exclusive alternatives:

$$X_1 + X_2 \leq 1$$

Investment *d* is conditional on the realization of investment a, that is, d can only be carried out if *a* has been done. These variables are known as contingent decisions, which are decisions that depend upon previous decisions.

$$X_4 \leq X_1$$

The increase of the productive capacity should not exceed the maximum sales forecast:

$$10000\,X_1 + 15000\,X_2 + 5000\,X_3 + 2000\,X_4 \leq 16000$$

5.3.2. SETUP COST PROBLEM

A company can manufacture four products on a production line that goes through three different departments. Table 5.3 shows the manpower/hour needed in each departments per one thousand product units, as well as the availability of hours per month, the gross profit (sale price - variable cost) and the fixed cost of conditioning the production line for each product, i.e. setup costs. The company wants to program the production to maximize the benefits.

The linear programming approach does not work when we have setup costs. In linear programming all costs are considered to be variable costs, that is, proportional to the value of the variable, while in this case there is a fixed cost only when the variable value is positive, but zero when the variable value is also zero. It should be emphasized that the setup cost problem only arises when the fixed cost is charged if there is production and is not charged if the production level is zero. When the fixed cost is always charged, the continuous linear programming is valid. Why?

Table 5.3. Technical and economical data

Product	Manpower-hour needed per one thousand units of product			Gross profit €/unit	Setup Cost €
	Dep 1	Dep 2	Dep 3		
P1	160	120	50	80	200,000
P2	150	200	50	85	200,000
P3	100	180	50	98	90,000
P4	200	175	50	100	150,000
Availability of manpower-hours/ month	4,000	4,800	1,600		

Let´s see the formulation of an appropriate mixed integer programming model to solve the previous problem. We define the **variables** P_1, P_2, P_3 and P_4 to be the size of each production batch in thousands of units.

$$P_j \geq 0 \quad for\ j = 1, 2, 3, 4$$

The profit of producing P_1 is

$$B_1 = 80{,}000\ P_1 - 200{,}000 \quad \text{for } P_1 > 0 \text{ and}$$

$$B_1 = 0 \text{ for } P_1 = 0$$

The profits of the other three products present a similar structure. To represent this by means of a linear function we define some binary variables Y_j which have a value of 1 when $P_j > 0$ and zero when $P_j = 0$.

The **objective function** is as follows:

$$\mathbf{MAX}\ 80{,}000\ P_1 + 85{,}000\ P_2 + 98{,}000\ P_3 + 100{,}000\ P_4$$

$$-\ 200{,}000\ Y_1 - 200{,}000\ Y_2 - 90{,}000\ Y_3 - 150{,}000\ Y_4$$

The **constraints** will refer to availability of man-hours in the three departments and they will have to guarantee that the setup cost is charged when there is production.

$$Dep1:\ 160\ P_1 + 150\ P_2 + 100\ P_3 + 200\ P_4 \leq 4{,}000$$

$$Dep2:\ 120\ P_1 + 200\ P_2 + 180\ P_3 + 175\ P_4 \leq 4{,}800$$

$$Dep3:\ 50\ P_1 + 50\ P_2 + 50\ P_3 + 50\ P_4 \leq 1{,}600$$

$P_1 \leq 100\ Y_1$

$P_2 \leq 100\ Y_2$

$P_3 \leq 100\ Y_3$

$P_4 \leq 100\ Y_4$

The last four constraints ensure that the setup cost is considered whenever the corresponding level of production is positive. The coefficient of the binary variables has to be a number large enough not to limit the level of activity of the P_j, in this case 100 is big enough to not limit the production values.

5.3.3. SITE SELECTION OF INDUSTRIES AND SERVICES

Another important application of mixed integer programming is found in problems where the objective is to determine the location and optimal size of a series of factories that produce high consumption goods. The demand and their clients' location are known.

The model for a product and one period could be the following:

Coefficients

$b_1, b_2..., b_n$: known demands for n clients (j=1, 2 …n)

$a_1, a_2..., a_m$: production capacities to install in m factories (i=1, 2 …m)

$f_1, f_2..., f_m$: construction cost of the factories (i=1, 2 …m)

C_{ij}: transport cost of a unit of goods from the i-th factory to the j-th client

Variables

X_{ij}: Number of transported units from factory i to client j

y_i: Binary variables, 1 indicates that the factory i is built and 0 otherwise for i = 1,2,...,m

Objective Function

The objective consists of minimizing the total cost (fixed cost of construction of the factories plus the distribution variable costs) satisfying the demand.

$$MIN \sum_{i=1}^{m} \left(f_i y_i + \sum_{j=1}^{n} C_{ij} X_{ij} \right)$$

Constraints

Clients demand

$$\sum_{i=1}^{m} X_{ij} \geq b_j \quad j=1, 2\ldots n$$

Production capacity of the factories

$$\sum_{j=1}^{n} X_{ij} \leq a_i y_i \quad i=1, 2\ldots m$$

This type of model has been applied to problems such as location of milk pasteurization centers, slaughterhouses, feed warehouses, waste treatment plants, etc...

5.3.4. A DISTRIBUTION PROBLEM WITH NONLINEAR COSTS

In this section we present a real problem of decision-making and mixed integer programming model that reduces the distribution costs of the company (Maroto, C.; Aliaga, S. and A. Torres, 2000).

The operation process of the company is represented in Figure 5.3. The company manages the broiler production process providing farmers with one-day-old chickens, feed and other resources such as technical and sanitary assistance. It is also responsible for collecting and marketing the fattened chicken. Broiler farmers provide the labor and equipped warehouses, and they are paid on the basis of average costs of breeding and production results. Given the economic importance of feed costs, the company is thinking of reducing transport costs from the factory to the farms. The distribution of shipments was carried out with subjective criteria and manually. Therefore, the company was also interested in **planning shipments of feed by objective criteria, minimizing costs.**

The company is responsible for estimating the feed required by each farm and placing orders to the feed factory. This offers prices with a cost structure shown in Table 5.4. As shown in the table, transport prices depend both on the weight of the order and the distance from the factory to the farm. Thus, the company is facing a **nonlinear transportation cost structure**. The previous method used by the company was to assess the needs for each type of feed on each farm and place as many large orders as possible and the difference with another order. The company is aware that these remaining orders increase the total cost.

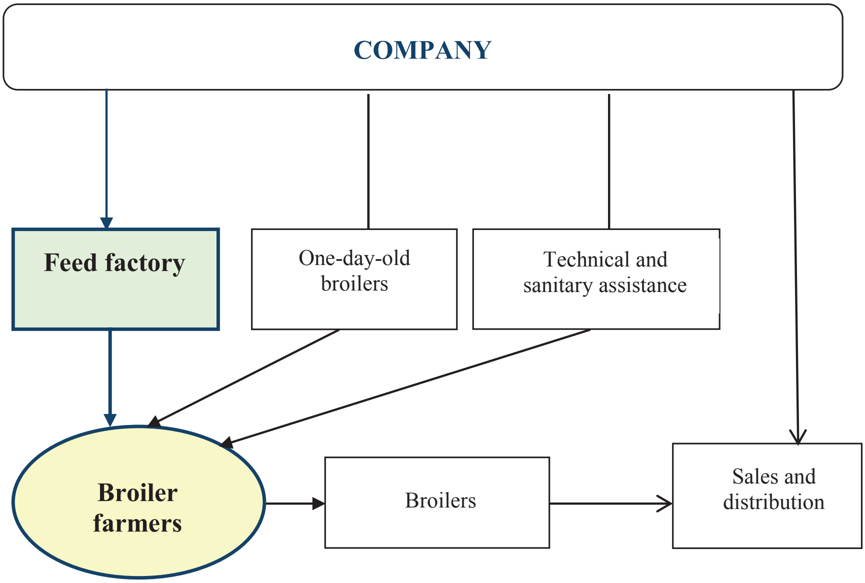


Figure 5.3. Operation process of the company

Table 5.4. Feed transport costs structure

Distance Km	20-24 Tons €/Tons	16-19.9 Tons €/Tons	12-15.9 Tons €/Tons	8-11.9 Tons €/Tons	4-7.9 Tons €/Tons
0-10	1.94	1.95	1.99	2.36	3.25
10.1-15	2.20	2.28	2.36	2.74	3.67
…					
100.1-105	6.94	8.17	9.02	9.30	11.06
…					
340.1-345	19.57	23.87	26.80	27.07	30.68
345.1-350	18.83	24.19	27.17	27.44	31.10

To solve this problem we proposed the following integer programming model. There are many models, at least one for each farm and each batch of broilers. All models have the same structure. Firstly we define the parameters which will have a specific value in each model.

Parameters

The feed orders depend on three factors: the capacity of the silos installed on the farms, the access road to them and the feed intake. The latter is based on the number of chickens, mortality, age, sex, season and type of warehouse. Large orders of 24 tons, which are the ones with lower cost, cannot always be considered if the size of the silo of the farm is smaller or if the accesses prevent the passage of trucks of that size.

The problem arises because transport costs are nonlinear, they depend on the **distance from the farm to the factory**, and on the **shipping amount**. Thus, for a given farm the most economical rate is that corresponding to 20-24 ton shipments (T1 € / ton). The second most economical rate is the shipping fee from 16 to 19.99 tons (T2), the third from 12 to 15.99 tons (T3), the fourth from 8 to 11.99 tons (T4) and the fifth, and most expensive, is the fee for the smallest shipping, from 4 to 7.99 tons (T5). We can state that the cost is nonlinear, but is constant within each of the five intervals.

Three types of feed must be provided throughout the animals' growth process and the **total amount** in breeding (P_1, P_2 and P_3) depends on the number and sex of the chicks, as well as other factors such as race, season and type facilities. When the company planned the deliveries manually they made all possible shipments of a larger size, generally leaving a small residue of shipping in tons, but with a very high unit cost. The model should minimize the cost of transportation of P_1, P_2 and P_3 tons of feed to a particular farm.

Another parameter affecting the order quantity is what we have called **CS_i** for i = 1, 2, 3, which is the **total capacity of the silo except the safety margin for the feed P_i**. Finally, **Ac is the maximum truck load** in terms of the **access roads to the farm**.

1. Variables

We defined two types of variables, continuous variables X_{ijk} representing tons of feed i (i = 1, 2, 3) at the rate j (1, 2 ... 5) in the shipping k (1, 2 ... K) necessary for broiler production on a given farm. K is estimated as the maximum number of shipping for each type of feed needed to supply the larger farm, during each period of consumption. The other type of variables is binary Y_{ijk}, which will take value 1 if the rate j is used in shipping k for feed type i and 0 otherwise. These binary variables are necessary due to the nonlinear structure of transport costs and permit us to formulate this problem using linear functions.

X_{ijk} tons of feed i=1, 2, 3 at rate j=1,2,…5 in the shipping k = 1,2…K

Y_{ijk} *(0, 1)* binary, 1 indicates that the rate j is used in the shipping k for the type of feed i and 0 otherwise.

2. Objective Function

The objective function consists of minimizing the transport cost of the feed needed for raising chickens on a specific farm and is given by the cost of all shipments to the farm. All 20-24 tons orders will be charged at price T_1, 16 to 19.9 tons at price T_2, 12 to 15.9 tons at price T_3, 8 to 11.9 at price T_4 and the orders between 4 and 7.9 tons at price T_5 which is the most expensive. The $\mathbf{T_j}$ are **fixed for each farm** and vary from one to another depending on the distance to the factory.

$$MinZ = T_1\sum_{i=1}^{3}\sum_{k=1}^{K}X_{i1k} + T_2\sum_{i=1}^{3}\sum_{k=1}^{K}X_{i2k} + T_3\sum_{i=1}^{3}\sum_{k=1}^{K}X_{i3k} + T_4\sum_{i=1}^{3}\sum_{k=1}^{K}X_{i4k} + T_5\sum_{i=1}^{3}\sum_{k=1}^{K}X_{i5k}$$

3. Constraints

Demand for each type of feed: we have a restriction for each type of feed that indicates that the sum of all the sent amounts must be equal to the total required.

$$\sum_{j=1}^{5}\sum_{k=1}^{K}X_{1jk} = P_1$$

$$\sum_{j=1}^{5}\sum_{k=1}^{K}X_{2jk} = P_2$$

$$\sum_{j=1}^{5}\sum_{k=1}^{K}X_{3jk} = P_3$$

Silo capacity: no order may exceed the capacity of the silo on the farm minus the estimated safety margin (CS_i). The safety margin varies depending on the number of broilers and their period of growth.

$$X_{ijk} \leq CS_i \quad \text{for i=1,2,3; j=1,…5; K=1,…K}$$

Truck access: the size of the orders is limited by the tonnage of the trucks that can access the farm.

$$X_{ijk} \leq Ac \quad \text{for i=1,2,3; j=1,…5; K=1,…K}$$

Price list (nonlinear costs): the following restrictions represent the structure of transport costs that the company has to pay to the feed factory. Thus, the first constraint indicates that all orders of feed i with the cheapest rate T_1 will have between 20 and 24 tons. There will be as many constraint groups of this type as there are types of feed and possible deliveries.

$$20\,Y_{i1k} \le X_{i1k} \le 24\,Y_{i1k}$$
$$16\,Y_{i2k} \le X_{i2k} \le 19.9\,Y_{i2k}$$
$$12\,Y_{i3k} \le X_{i3k} \le 15.9\,Y_{i3k}$$
$$8\,Y_{i4k} \le X_{i4k} \le 11.9\,Y_{i4k}$$
$$4\,Y_{i5k} \le X_{i5k} \le 7.9\,Y_{i5k}$$

All the constraints of the price tariff for $i = 1, 2, 3$ and $k = 1, 2,\ldots K$

The solution to this model gives the number of orders that we should place to the feed factory and the measured quantity in tons for each batch of chickens and farm, minimizing the cost, which is the largest production cost of the company and improving its decision-making. This model not only reduces the costs of their activities, but it also reacts appropriately to any unforeseen needs such as reduction of necessities due to animal death on hot days, electricity failures, diseases, etc. by simply re-running the model with the updated data.

Let us consider a simple example of a similar problem. A company needs 85 tons of product 1 and 90 tons of product 2 for next month. The provider has a price that depends on the size of the order. Determine the number of orders to be placed in order to minimize the cost of the product.

Table 5.5. Transport cost´s structure

Amount of order in tons	Cost of Transport Euros/ton
16-20	8
12-15.9	12
5-11.9	16

The **variables** are X_{ijk} = tons of product i at price j in shipment k;

The **objective function**

MINIMIZE $8*(X_{111} + X_{112} + X_{113} + X_{114} + X_{115} + X_{211} + X_{212} + X_{213} + X_{214} + X_{215}) +$

$12*(X_{121} + X_{122} + X_{123} + X_{124} + X_{221} + X_{222} + X_{223} + X_{224}) +$

$16*(X_{131} + X_{132} + X_{133} + X_{134} + X_{231} + X_{232} + X_{233} + X_{234});$

Constraints:

Demand of product P_1

$X_{111} + X_{112} + X_{113} + X_{114} + X_{115} + X_{121} + X_{122} + X_{123} + X_{124} + X_{131} + X_{132} + X_{133} + X_{134} = 85;$

Demand of product P_2

$X_{211} + X_{212} + X_{213} + X_{214} + X_{215} + X_{221} + X_{222} + X_{223} + X_{224} + X_{231} + X_{232} + X_{233} + X_{234} = 90;$

Currently we can solve this nonlinear cost problem by substituting equations linking the amount of the product with the corresponding binary variable using a type of variable known in LINGO as semi-continuous. For example, to indicate that if the variable X_{111} has a nonzero value in the optimal solution then this is between 16 and 20, we add to the model the following

@SEMIC (16, X_{111}, 20)

In this case LINGO generates the necessary binary variables and constraints to take this situation into account. However, in Excel this is not possible and you have to enter all the restrictions, define the binary variables and take into account that in Excel we have to put all variables on the left-hand-side and in the right-hand-side of the constraints only the independent term.

5.3.5. A PROBLEM OF TRANSPORT ROUTES

Case study 3 of chapter 4 is a problem of feed manufacturing and distribution for a multinational company in the food industry which has one of its factories in Valencia. In the aforementioned case a linear programming model was formulated to minimize manufacturing costs. In this section we will explain an integer programming model to solve the feed distribution problem to customers, which differs from the theoretical route models presented in Operations Research books.

In short, the company manufactures about 150 animal feed products and distributes 800,000 kg per day to an average of 70 clients. The company has a portfolio of 700 clients, whose distance from the factory varies from 15 to 450 km. The company has hired a fleet of trucks with capacities of between 12 and 24 tons, with compartments of 4 tons. Therefore, each delivery route can visit at most six customers. The person responsible for calculating the routes dedicates six hours per day to this activity. In addition, between 5-30 rush orders are received every day, which means that the routes cannot be recalculated to include these orders. The cost of the routes is a complex function dependent on the distance to the last customer served and the truck loading, which has a minimum cost even if there is no transport of goods. For example, if a 24-ton truck carries 20 tons, the company pays as if 23 tons were transported.

Theoretical route models do not resolve this real problem because, amongst other reasons, the objective function does not seek to minimize the distance driven, but rather the cost according to the company's contract with the carriers.

To solve this problem we first generate all feasible routes by an implicit enumeration algorithm. *A priori* one might think that the high number of clients to be served will give us astronomical figures. However, taking into account all of the real problem limitations only a small percentage of all theoretical routes are feasible. Thus, a 24-ton truck cannot visit more than 6 clients, a truck cannot fulfill orders totalling more than its capacity, routes cannot be more than 450 km and certain trucks cannot serve some customers due to restrictions on the accesses such as bridges, tunnels or narrow roads. The implicit enumeration method gives us all feasible routes and for each customer to be visited, the total distance driven, the route cost and the type of truck used.

The integer programming model that selects routes that minimize the cost of feed distribution is as follows.

Parameters:

N **=** Number of possible routes

C= Number of orders

$Cost_n$ = Cost of route n

PP_{nc} is 1 if the route n serves the order of client c and 0 otherwise

Where n = [1, 2…N] and c = [1…C]

Variables:

X_n binary, are 1 if we take route n and 0 otherwise

Objective Function:

$$Min \sum_{n=1}^{N} Cost_n X_n$$

Constraints:

$$\sum_{n=1}^{N} PP_{ni} X_n = 1, \forall i \in [1,...,C]$$

LINGO solves this model in less than 1 minute for regular size models and in less than five minutes for models with more than 35,000 binary variables. By using the model the company has reduced the cost of transportation by between 9-11%. The distance driven by the trucks has also decreased by between 7-12% thereby also reducing pollution due to transport of goods.

5.3.6. OTHER FORMULATION POSSIBILITIES WITH BINARY VARIABLES

1. Either one constraints

This situation is presented when we have to choose between two constraints in such a way that at least one of them is fulfilled, but not necessarily both, for example

$$3\,X_1 + 2\,X_2 \leq 18$$

or

$$X_1 + 4\,X_2 \leq 16$$

This condition has to be reformulated as a mixed IP model in which all constraints specified have to be met. This can be achieved by adding a large M number to the right-hand-side, as any solution that satisfies the other constraints of the problem will automatically meet this. This formulation entails that the set of feasible solutions of the complete problem be a bound set and that M is large enough not to eliminate any of those feasible solutions.

This formulation is equivalent to:

$$3\,X_1 + 2\,X_2 \leq 18 + Y\,M$$

$$X_1 + 4\,X_2 \leq 16 + (1 - Y)\,M$$

Y being an auxiliary binary variable (0, 1). If two variables were used instead of a binary variable, the condition $Y_1 + Y_2 = 1$ would have to be added.

K out of N constraints (K < N) have to be met

The objective is to choose which combination of K constraints allows the objective function to have the best possible value. This case is a direct generalization of the previous case for which K = 1 and N = 2.

$$f_1\,(X_1, X_2 \ldots X_n) \leq d_1$$

$$f_2\,(X_1, X_2 \ldots X_n) \leq d_2$$

$$.$$

$$f_N\,(X_1, X_2 \ldots X_n) \leq d_N$$

The procedure would be the following:

$$f_1 (X_1, X_2,\dots X_n) \leq d_1 + M\, Y1$$
$$f_2 (X_1,X_2,\dots X_n) \leq d_2 + M\, Y2$$
$$.$$
$$f_N (X_1,X_2,\dots X_n) \leq d_N + M\, Y_N$$

and also

$$\sum_{i=1}^{N} Y_i = N - K$$

2. Functions with N possible values

$$f (X_1,X_2,\dots, X_n) = d_1, \text{ or } d_2,\dots \text{ or } d_N$$

The procedure would be the following: $f(X_1,\dots X_n) = \sum_{i=1}^{N} d_i Y_i$

$$\sum_{i=1}^{N} Y_i = 1$$

where Y_i is a (0,1) variable, i.e. auxiliary binary variable.

There are other types of problems that can be solved by means of integer programming, where this enters in competition with other possible solving techniques. It happens thus, for example, with those of "Knapsack type" and the scheduling problems.

There are two versions of the **Knapsack problem**. In the first one, a certain space is provided with certain volume or capacity that should be filled with valuable items and volume or specified capacity. The problem consists of filling that space with the most valuable group of objects, without exceeding the physical limits of this space. The second version consists of dividing an object into several portions with different values and trying to find the division with the greatest value. When the number of variables of a problem with this structure is small, the dynamic programming is quite appropriate, but if the number of variables is higher than 15 or 20, dynamic programming competes unfavourably with the Branch-and-Bound algorithms. Knapsack problem are used to solve investment problems and problems of determination of the size of a fleet of vehicles.

With regard to **scheduling problems**, they refer to situations in which it is necessary to provide the sequence of a certain number of jobs in a series of machines, with the purpose of minimizing the cost or the time. This type of problems arises, for example, in tile and textile companies that work on demand.

Finally, note that the **optimization models of transport, assignment and optimization networks** are integer programming models. However, due to their special structure they have more efficient specific algorithms. In certain situations the alternative to integer programming is the use of metaheuristic algorithms, as in the case of obtaining a sequence of activities so that the total time is minimum and keeping in mind the limitations of resources (chapter 9).

5.4. INTEGER PROGRAMMING TECHNIQUES: BRANCH-AND-BOUND ALGORITHMS

5.4.1. INTRODUCTION

Gomory was the author of the first cutting-plane algorithm to solve integer programming models (1958) and soon afterwards, in 1960, Land and Doig presented another more promising method than the algorithms based on cutting planes. This method was the first one of what is known today as the "Branch-and-Bound" algorithm. Then, in the 80s ***Branch-and-Cut*** algorithms were developed, which are nothing more than an integration of both types of methods.

Gomory's algorithm consists of solving the problem without considering the constraints of the integer variables and if the optimal solution is not integer it adds constraints that reduces the feasible region of the linear programming model, without excluding any integer solution.

The Branch-and-Bound algorithms are based on a similar idea, in the sense that in the first place they solve the continuous linear programming model and later they add constraints. The main difference is that the effect of such constraints is now to divide the feasible region in two. When making this division, no feasible integer solution is eliminated. These two subproblems are solved and if we find an integer solution, we can see the value of the objective function. If this is better than the value of the objective function of the other subproblem, we have finished and that will be the integer optimal solution. Otherwise, each one of the subproblems is divided into other two and so forth.

How are these constraints generated? Let us see this with an example analyzing the algorithm procedure in a graphical way.

5.4.2. GRAPHICAL SOLUTION

Let the integer programming model P_0:

$$\text{MAX} \quad 3 X_1 + 4 X_2$$

With the following constraints:

$$2 X_1 + X_2 \leq 6$$

$$2 X_1 + 3 X_2 \leq 9$$

$$X_1, X_2 \geq 0 \text{ and integers}$$

In the first step, the corresponding linear programming model is solved without considering the constraints that X_1 and X_2 have to be integer. If the solution has integer values it would be the optimal one. When it is not so, as in this case, since the continuous solution is $X_1 = 2.25$, $X_2 = 1.5$ and $Z = 12.75$, one of the variables that should be integer but is not, is taken, for example X_1, and two new problems are generated.

Each one of the subproblems generated is the previous one plus one constraint that forces this variable to have a minimum value of the higher integer or a maximum of the lower integer. These two new subproblems are P_1 and P_2.

P_1: Max $3 X_1 + 4 X_2$

$2 X_1 + X_2 \leq 6$
$2 X_1 + 3 X_2 \leq 9$

$\mathbf{X_1 \geq 3}$

P_2: Max $3 X_1 + 4 X_2$

$2 X_1 + X_2 \leq 6$
$2 X_1 + 3 X_2 \leq 9$

$\mathbf{X_1 \leq 2}$

In Figure 5.4 we can see the graphical solution of problem P_0 and that of the sub problems P_1 and P_2. In the case of sub problem P_1 the feasible region is formed by only one point, in which the variables X_1 and X_2 are integer. However, we cannot say yet that this is the optimal solution, since the solution of P_2, although not integer has a better value for the objective function. What we can guarantee is that P_1 provides a lower bound for the value of the objective function of the optimal integer solution.

Therefore, the next step is to divide P_2 into two new problems: P_3 and P_4. Since only X_2 is not integer, the only constraints to add are those that force X_2 to take the higher or lower integer value $X_2 \geq 2$ and $X_2 \leq 1$. Figure 5.4 shows the graphical solution of these models and we can see that P_4 has an integer solution with objective function equal to 10 and better than P_1. Thus, 10 will now be the lower bound for the value of Z of the integer programming model.

We have to continue by dividing P_3 into two other models, P_5 and P_6, since the value of Z is better than the lower bound. Also in figure 5.4 (continuation) we can see that the model P_5 does not have a feasible solution as there is no point that satisfies all of the constraints. P_6 has a solution indeed, and as it is not an integer we have to create another two new linear programming models: P_7 and P_8. In Figure 5.4 (continuation) we can see that in the first of them the feasible region is formed by only one point, where the levels of activity of the variables are integer. In the second, P8, the optimal solution is also integer. Subproblem P_7 is the one with the best value of Z of the last integer solutions, therefore it is the optimal integer solution of problem P_0. The optimal solution is X1 = 0, X2 = 3 and Z = 12.

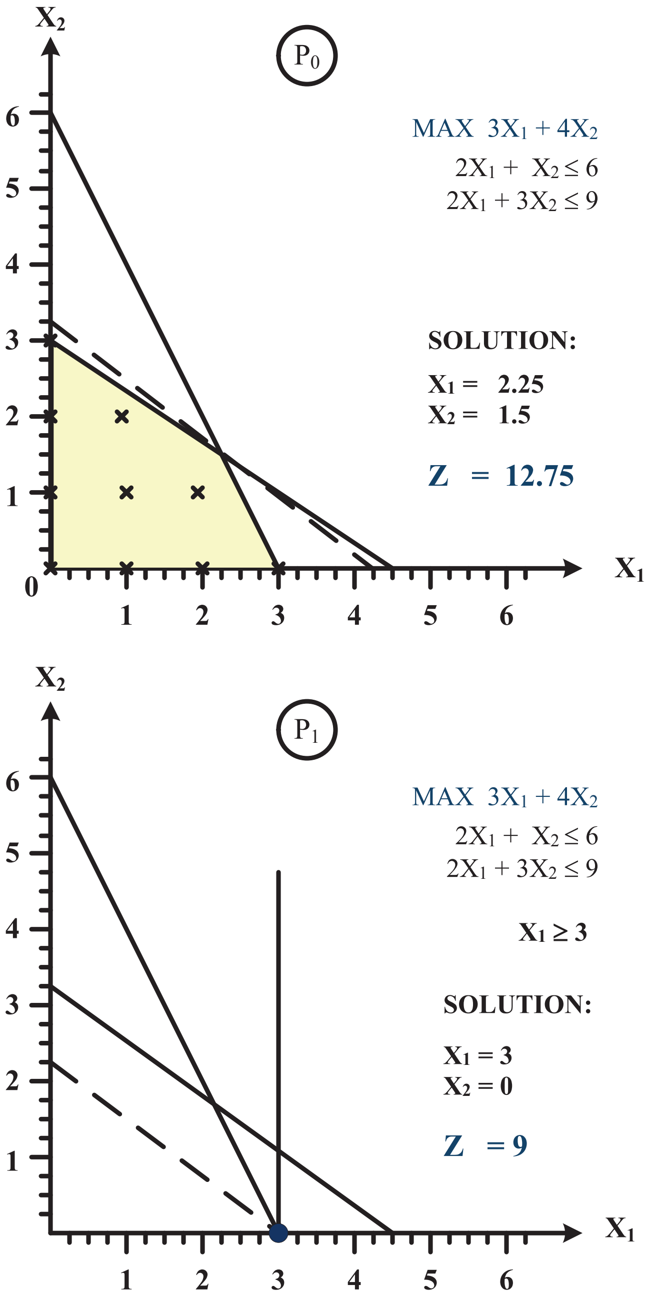
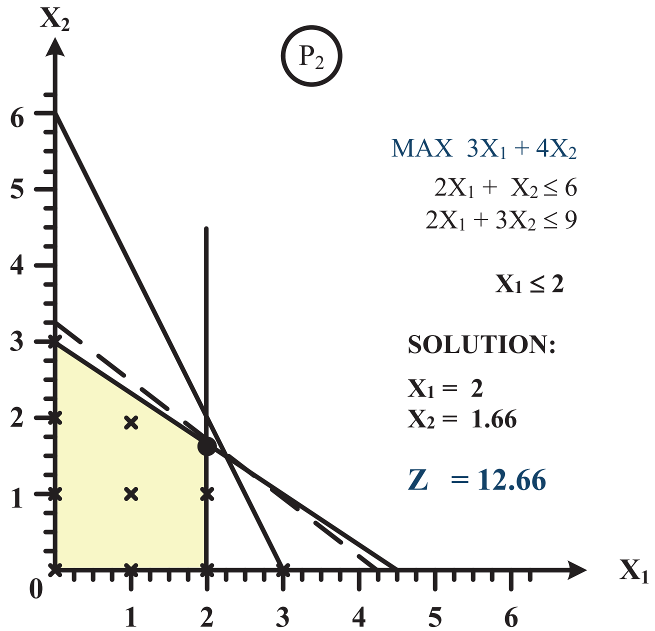


Figure 5.4. Graphical solution of the problem



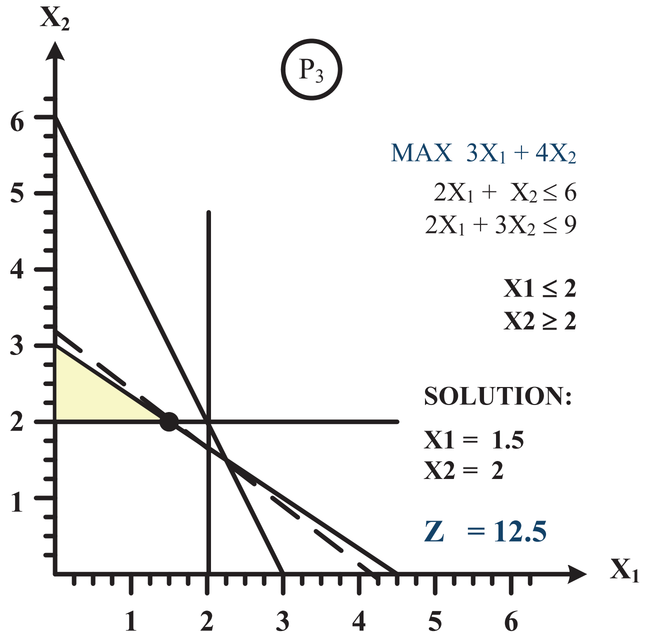


Figure 5.4. Graphical solution of the problem (continuation)

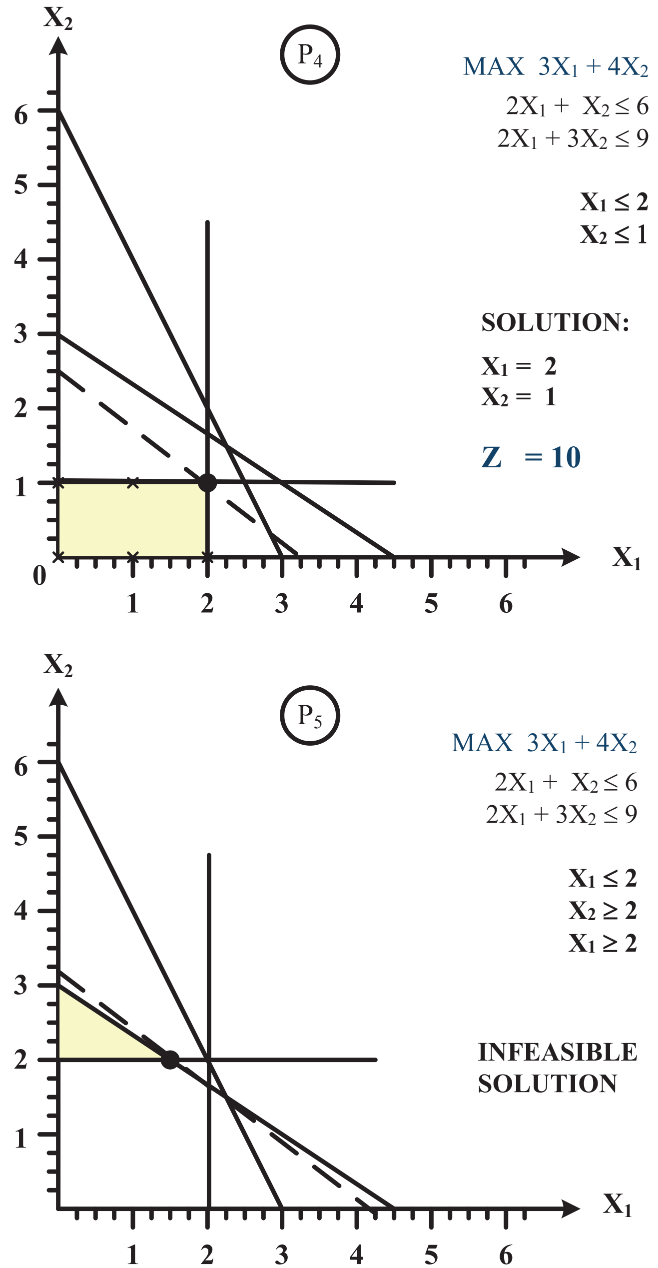


Figure 5.4. Graphical solution of the problem (continuation)

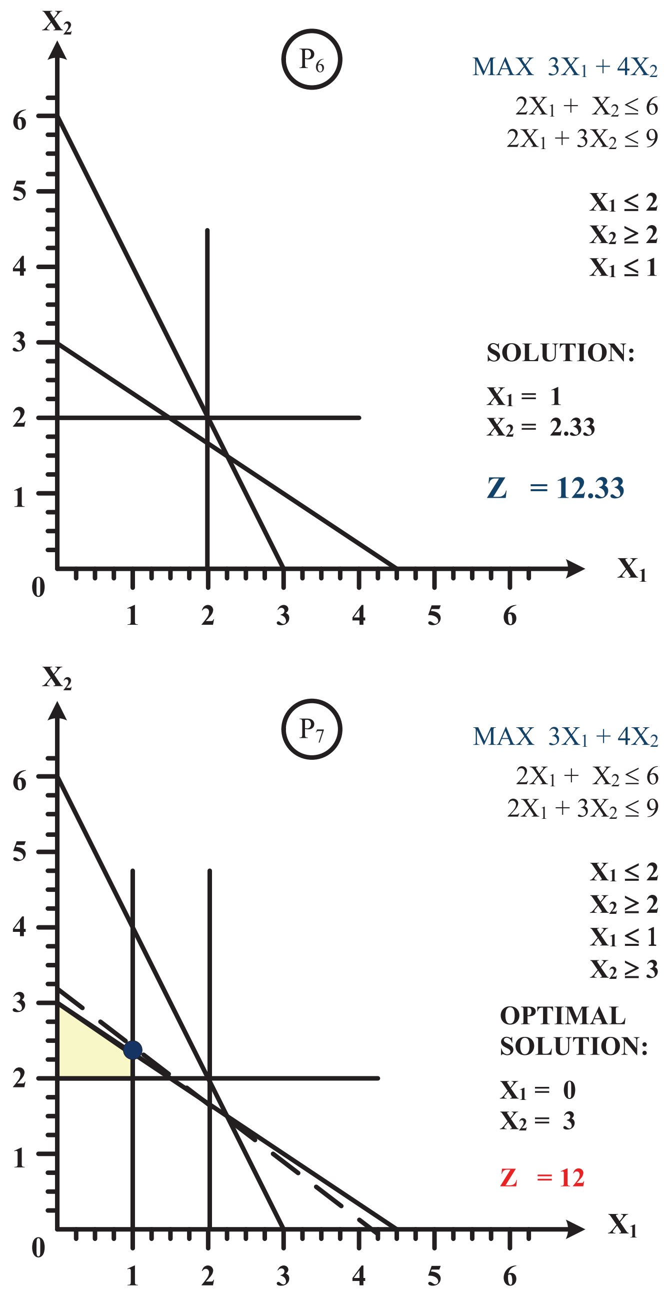


Figure 5.4. Graphical solution of the problem (continuation)

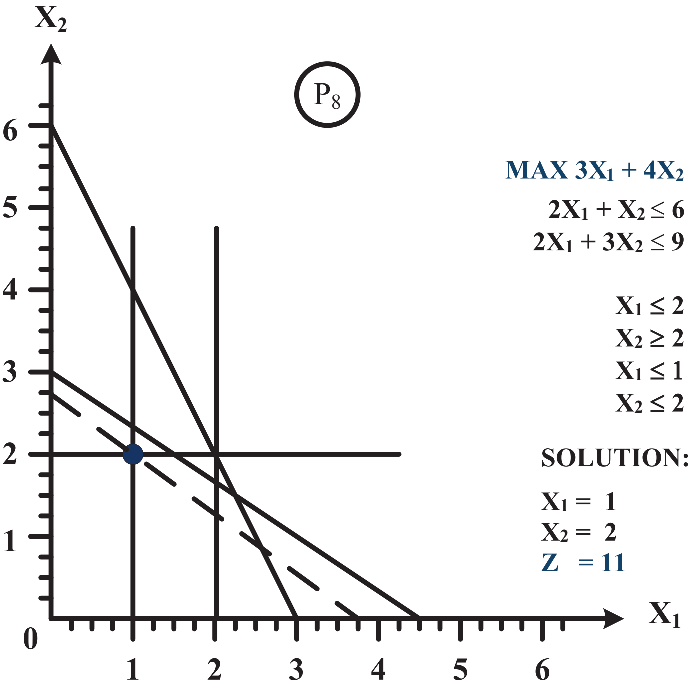


Figure 5.4. Graphical solution of the problem (continuation)

5.4.3. SELECTION CRITERIA OF THE NODE

In the previous sections we have seen the procedure of the Branch-and-Bound algorithm for a simple integer problem. We have just subdivided the problem in two subproblems and bound variables. In the upper part of Figure 5.5 we present the process followed in our example in tree form. What nodes have we chosen to make the following branching? The selection has been to continue the branch with the best value of the objective function. The possible criteria in this respect are:

1**. Technique of the best bound**

It consists of choosing the node with the best continuous optimal value, seeking to find optimal feasible solutions. However, in large problems, this approach usually leads to having to store too many data and it generates computer memory problems.

2. Technique of the most recent bound

This criterion consists of choosing the node that was created more recently. Thus, you advance quicker down the tree and, although you have to examine more nodes than with the previous approach, it is easier to eliminate them during the process.

In both cases we should keep the node with the best integer solution obtained so far. This allows us to eliminate all those that have a worse value of the objective function and even those with a smaller better percentage in which the variables are not integer, since if you continue along these nodes it will imply new losses in Z. If we operate in this way we will not have a guarantee of finding the best solution, but we will find a "quasi-optimal" solution.

In the lower part of Figure 5.5 we represent the tree obtained when applying the approach of the most recent node to the same example. Compare this tree with that obtained when applying the best bound technique.

Once we are in the node for which we will continue branching: Which variable of the ones that must be integer and have no integer value in the solution, do we bound? We should select the variables with the greatest impact on the model. For example, to make a large investment or not will be a more important variable than the number of workers needed. Observe in Figure 5.6 the tree that is obtained from the above example if we start by delimiting the variable X_2. LINGO has options where the user can choose the node selection strategy and permits the prioritisation of the binary variables, as explained in annex 2.

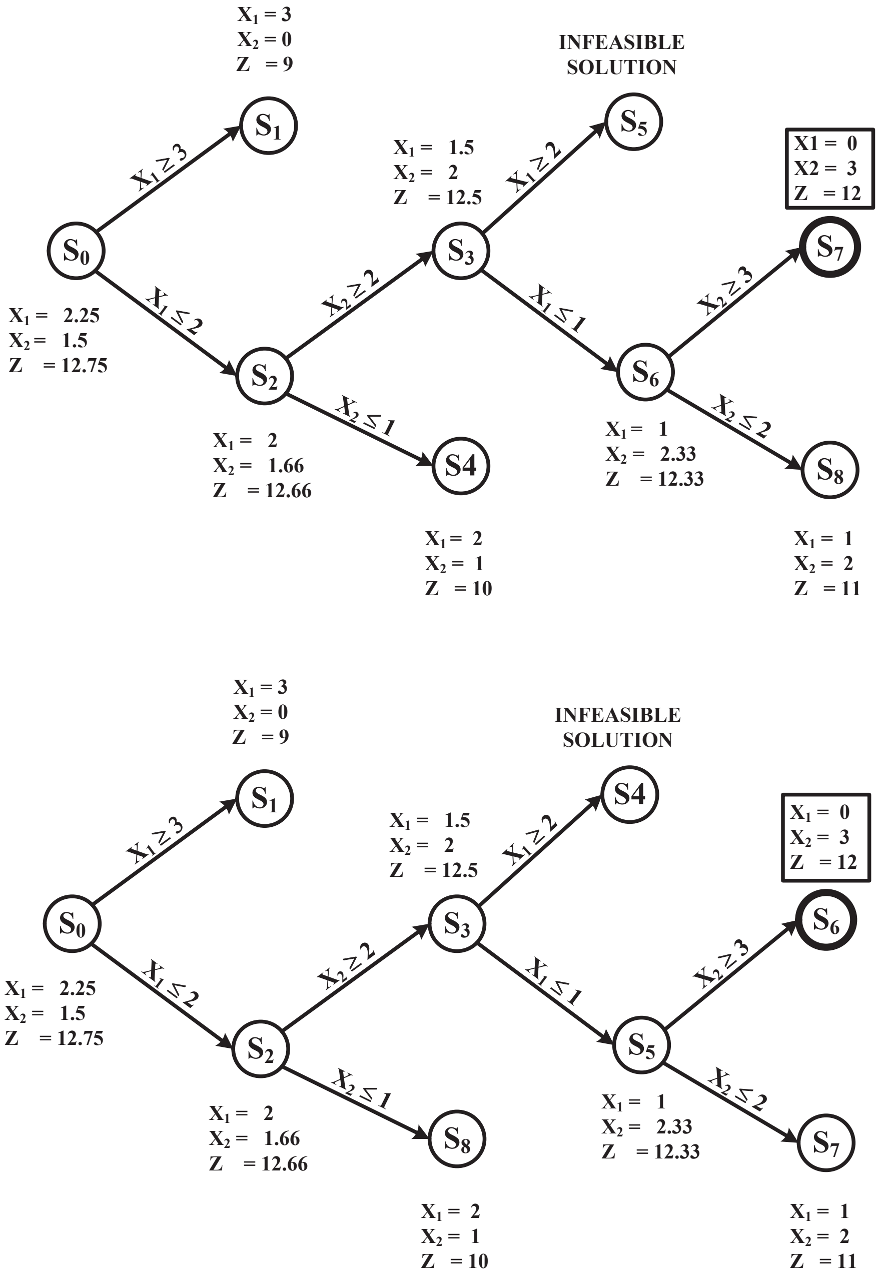


Figure 5.5. Selection criteria of the node:
1. Best bound. 2. Most recent bound

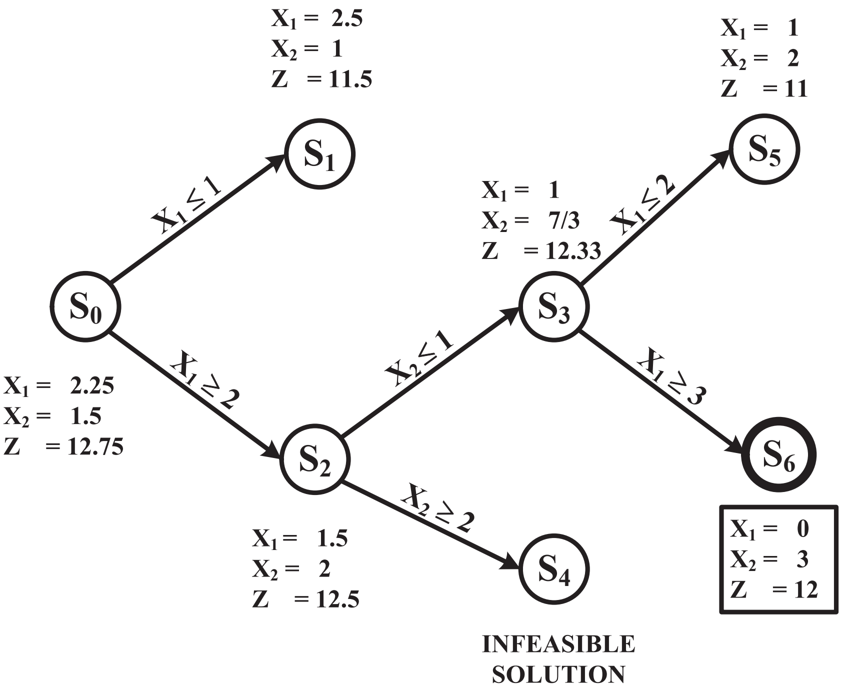


Figure 5.6. Solution tree branching variable X_2

5.5. BRANCH-AND-BOUND TECHNIQUES AND OPTIMIZATION SOFTWARE

The optimization software uses the Branch-and-Bound algorithm as seen in the previous section. The latest tendencies in the application of this type of algorithm consist of adding constraints (cuts) to reduce the set of feasible solutions of the corresponding linear programming model, without eliminating any integer solution. For this reason, optimization software currently uses the *branch and cut methods* in a pre-processing phase.

This automatic pre-processing consists of reformulating the model in order to solve it easily without eliminating any integer feasible solutions. These reviews can be grouped into three categories:

1. Set variables: Identify binary variables that may bind themselves to one of its two values. For example, if we have the restriction $3Y \leq 2$ where Y is binary, it is obvious that in the optimal solution Y can only be zero.

2. Remove redundant constraints.

3. Reduce the continuous feasible region without eliminating integer solutions using cutting planes.

LINGO by default uses the *branch and cut* method, but you can limit the number of constraints to be added if you think that the process does not improve in that way. In the real case explained in section 5.3.4 we found that we obtained the best results by selecting the option of cuts of "Gomory and GUB". LINGO has another option called **Hurdle** that allows us to enter the objective function value of a previously known integer solution. This means that the program can shorten the search process, only continuing branches whose objective function is better than the indicated value.

In integer programming the time required to solve the problem depends primarily on the number of integer variables rather than the number of functional constraints of the problem. It can even be the case that when it is more restricted it is easier to solve. The application of branch and bound algorithms on large problems can be computationally infeasible due to the required computing time. We can also meet with limitations in memory storage. Therefore, the heuristic algorithms constitute a reasonable alternative if the above methods are too expensive.

Some heuristic algorithms are merely modifications of exact algorithms that allow us to reduce time and memory requirements needed at the expense of not guaranteeing an optimal solution. Thus, we can interrupt the algorithm execution and keep the best integer solution obtained so far. Another possibility, common in integer programming software, is to indicate to not analyze the nodes whose objective function is worse than a certain value or worse than a certain percentage of an integer solution. LINGO has an option that permits us to specify a number r between 0 and 1, which is the tolerance used by the branch and bound algorithm in the process of searching the integer solution. If the best integer solution has Z as the value of the objective function, the software just continues the search for branches that are at least r*Z better. In this case we do not necessarily find the best solution, but accelerate the resolution process for large models, knowing the maximum that we have lost. Other heuristic techniques to solve integer programming problems are very different to the branch and bound algorithms, such as genetic algorithms explained in Chapter 9. Another interesting option is **Relative Integrality Tolerance** that allows us to enter a number to check the integrity of the variables. X is considered integer if $|X-I| / |X|$ < tolerance, where I is the closest integer to X.

Finally, we note that the software uses the **simplex algorithm** to obtain the continuous solution and the **dual algorithm to reoptimize**. Obviously it is not necessary to solve the whole problem again on each node of the branch and bound algorithm. We can predict that future methods on this topic will be integrated with *constraint programming* techniques. At present, these techniques allow us to formulate models in a more compact manner and find feasible solutions more efficiently than integer programming. However, integer programming is a much more powerful technique for finding the optimal solution amongst all of the feasible ones than constraint programming.

5.6. SUMMARY

In this chapter we have addressed specific integer programming models, such as capital budgeting, transport routes and the site selection of industries where binary variables are used as decision variables and models with nonlinear cost functions where the binary variables, in this case auxiliary variables, allow us to represent the actual costs in linear format.

Regarding solving methods, the general algorithms of integer programming that professional software currently uses have been explained, the branch-and-bound algorithms-, as well as possible ways to use them to overcome the difficulties that may arise. For example, we have seen the possibilities for node selection and prioritizing certain variables in the search process.

5.7. SELECTED REFERENCES

1. Daellenbach,H.G.; George,J.A.y D.C.McNickle (1987): *Introducción a las técnicas de investigación de operaciones*. CECSA.

2. Hillier, F.S. and Lieberman G.J. (2010): Introduction to Operations Research. Ninth Edition. McGraw-Hill.

3. Hillier, F.S and Hillier, M.S. (2011): Introduction to Management Science. A Modelling and Case Studies Approach with Spreadsheets. Fourth Edition. McGraw-Hill.

4. Maroto, C.; Aliaga, S. and A. Torres (2000): Optimización de Envíos de Pienso a Granjas de Broilers. *Boletín de la SEIO*. Vol. 16/4, pág.4-7.

5. Maroto, C.; Sánchez, L. y Robles, M. (2000): Formación de Médicos Residentes en un Servicio de Radiología: Planificación Óptima Mediante Modelos de Programación Entera. *Boletín de la SEIO*. Vol. 16/4, pág.2-4.

6. Pardo, L. et al. (1990): *Programación lineal entera. Aplicaciones prácticas en la empresa*. Díaz de Santos.

7. Ruiz, R.; Maroto, C. and J. Alcaraz (2004): A Decision Support System for a Real Vehicle Routing Problem. *European Journal of Operational Research* Vol. 153/3, 593-606.

8. Winston, W. L. and S.C. Albright (2012): Practical Management Science. Fourth Edition. South-Western. USA.

5.8. CASE STUDIES

CASE STUDY 1

1.1. Solve the following problem:

$$\text{MAX } 16\,X_1 + 22\,X_2 + 12\,X_3 + 8\,X_4$$

$$5\,X_1 + 7\,X_2 + 4\,X_3 + 3\,X_4 \leq 14$$

$$X_i = (0, 1) \text{ binary variables for } i=1, 2, 3, 4$$

Draw the resulting solution trees when applying the most recent bound technique and the best bound technique. Analyze the advantages and disadvantages of each criterion.

1.2. Solve the following integer programming model:

$$\text{MAX } 4\,X_1 + 3\,X_2 + X_3$$

$$3\,X_1 + 2\,X_2 + X_3 \leq 7$$
$$2\,X_1 + X_2 + 2\,X_3 \leq 11$$

$$X_1, X_2 \geq 0 \text{ and integer}$$

CASE STUDY 2. COVERING MODELS

The models presented in the problems below have a similar structure. In general, minimizing an objective function of the decision variables and constraints ensures that the selection of variables *covers* the requirements of the problem.

2.1. SHIFT SCHEDULING PROBLEM

Many industries have the problem of assigning a work schedule to their employees in order to cover some known service needs, in such a way that the number of workers is the minimum possible. To approach this problem we need to know the necessary manpower at each point of the day and the establishment of some types of shift work that comply with the legislation in force.

Supposing a working day of ten hours, in which each shift should complete 8 working hours. Table 5.6 shows the minimum number of employees in service in every hour and takes into consideration that the incorporation should be carried out at the beginning of every hour. The breaks, minimum of one hour, cannot be taken before having worked three hours nor after more than five.

Formulate and solve a model to determine the shifts that cover the needs and require the smallest number possible of workers. Add to the previous model the condition that it cannot have more than 2 working shifts.

Table 5.6. Number of employees required

HOURS	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18
Number of employees	20	40	60	80	60	50	40	50	60	30

2.2. PROBLEM OF OPTIMIZATION OF RAW MATERIAL CUTTING

A machine produces rolls of paper with a standard width of 180 inches. Some orders are received for at least 200 rolls of 80 inches wide, 120 rolls of 45 inches and 130 of 27 inches. Formulate and solve a linear programming model that allows us to decide how to cut the 180-inch-wide rolls to cover the orders with a minimum loss of paper.

2.3. ANOTHER COVERING PROBLEM

Channel 9 will broadcast the football game of the final of the King's Cup between Valencia and Real Madrid. It wants to know the minimum number of cameras that are needed in the stadium to cover all points of the field perfectly. The field has been divided into 20 sectors, which can be covered from 10 points as shown in Table 5.7. Formulate and solve a model to inform this decision-making problem.

Table 5. 7. Sectors covered by the cameras depending on their localization

Localization point	Sectors covered				
1	16	17	18	19	
2	17	18	19	20	
3	16	17	18	19	20
4	1	2	3	4	5
5	2	3	4	5	
6	7	8	9	10	
7	12	13	14	15	
8	6	7	8	9	
9	11	12	14	15	
10	7	8	9	13	14

CASE STUDY 3. CAPITAL BUDGETING

A company has € 1,000,000 to invest in 7 different funds. The company wants to maximize the total return on investment, taking into account the following conditions:

1. They will only invest in fund 2 if they invest in fund 1.
2. If they invest in fund 1 or 4 they do not invest in fund 6.
3. If they do not invest in fund 2 or 5, they will not invest in fund 7 either.
4. If they invest in a fund the amount invested in that fund must be between 5 and 40% of the total amount.

The return of the individual funds is shown in the following table 5.8.

Table 5.8. Funds return

	Fund 1	Fund 2	Fund 3	Fund 4	Fund 5	Fund 6	Fund 7
Return %	1	2,5	5,5	1,5	0,5	4,5	3,5

1. Formulate a linear programming model to maximize the total return on investment.

2. The company has an additional € 500,000 and wish to study whether to invest this whole amount in patriotic bonds, which have a return of 5%, or add it to the initial amount to be redistributed between the previous funds. Formulate a linear programming model to maximize the total return in this new situation.

CASE STUDY 4. LOCATION PROBLEM

An international consulting company wants to open two offices in Spain. Potential customers, both public and private, have been analyzed as well as the availability of qualified personnel and the communication facilities of different Spanish cities. Following a previous study four candidate cities: Madrid, Valencia, Bilbao and Seville have been selected.

Table 5.9. Average cost of serving a customer of an area (column) from the office located in the city indicated in the row

Offices	Centre	East	North	South
Madrid	2	5	6	7
Valencia	4	3	8	6
Bilbao	5	6	3	8
Sevilla	6	5	10	4
Number of clients	40	30	25	15

Formulate a model that allows the company to decide in which two towns to locate the offices and from which of them to serve customers located in the other two areas. Each office should provide customers with service in the area in which it is located and the area or areas assigned. The fixed cost of the office in Madrid is 35 m.u., 25 in Valencia, Bilbao 15 and 20 in Seville and the company's objective is to minimize the total costs.

CASE STUDY 5

A multinational company has decided to install four branches on four pieces of land that it has in regions A, B, C, D, E and F in the next four years. If two of these branches are installed in regions A and C the construction of the necessary facilities should be simultaneous. If in the first year, they also install a branch in region B, they will install one in region E.

If they choose to open branches in regions D and E, the facilities shall be constructed before the third year. Finally, the facilities in regions F and B cannot be installed during the same year.

The estimated costs in millions of Euros for the construction of the necessary facilities to locate the four new branches by region and year of construction is shown in Table 5.10.

We want to know the best location and the year of construction of the new centers that minimizes the cost, considering that management does not want to invest more than three million euros in the first year. Formulate an integer linear programming model that allows us to make the best decision.

Table 5.10. Construction costs of the branches in millions of Euros

Regions	Year 1	Year 2	Year 3	Year 4
A	1.5	1.3	1.8	1.6
B	1.4	1.5	2	2.5
C	1.9	2	2.2	2.3
D	1.4	1.5	1.5	1.6
E	1.3	1.4	1.8	1.5
F	1	1.3	1.6	1.8

CASE STUDY 6. A PROBLEM OF RENEWAL OF EQUIPMENT AND PRODUCTION PLANNING

An electric wire manufacturer is revising its machinery. Some of the current machines are almost worn out or obsolete, therefore the production costs are becoming unacceptably high. At the same time a change in the technology has caused an extension in the demand in terms of the wire type that this manufacturer produces. The current plant cannot cover the projected demand for the near future (Table 5.11). The following Figure shows the product flow and the machine types currently installed.

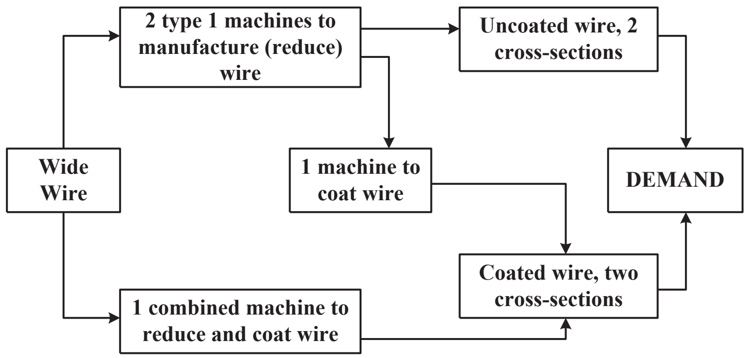


Figure 5.6. Existing plant and production flow

The coated wire can be manufactured by using either of two different processes. The first process manufactures the copper wire with the desired cross-section in a machine and it coats it in another machine. The second process uses a single machine to manufacture and coat the wire. While this second process is preferable because it implies less handling and waste, there is a certain demand for uncoated wire, therefore it is necessary to have some wire reduction facilities for uncoated wire.

Table 5.11. Planned demand of sales (Km)

Uncoated wire (Km)		Coated wire (Km)	
Num.1	Num.2	Num.1	Num.2
3,000	2,000	14,000	10,000

Table 5.12. Machine options and data

Data	Wire reduction machines		Coating wire machines		Combined machines
	Type 1 Existing	Type 2 New	Existing	Improve existing	New
Machine option	1	2	3	4	5
Capital costs in thousands (mu)	0	200	0	100	500
Operation costs per hour (mu)	5	7	8	8	12
Fixed cost per year in thousands (mu)	30	50	80	100	140
Production rate:					
Wire Num.1 m/h	1000	1500	1200	1600	1600
Wire Num.2 m/h	800	1400	1000	1300	1200
Product waste in %	2	2	3	3	3
Cost of waste in mu per 1,000 m	30	30	50	50	50

The company has several options, including the option of keeping and modifying some of the existing machines and acquire new machines. Table 5.12 shows these options together with the cost and the production data for each one of the possibilities. More than one machine can be acquired for options 2 and 5. None of the new or old machines has any resale value.

1. Formulate a model that provides the configuration of machines, as well as the proportion of annual time that each type of machine dedicates to produce the different types of wire. Suppose that the machines work 20 h/day, that is, 6,000 h per year.
2. Solve the model firstly by taking into consideration that the objective is to minimize the investment cost and secondly, assume that the objective is to achieve a configuration of the factory that minimizes the total operation costs.

3. Analyze the advantages and disadvantages of the two previous solutions and indicate which one you would implant if you were the decision-maker.

CASE STUDY 7. PLANNING FOR THE TRAINING OF MEDICAL SPECIALISTS

This case study presents a real problem consisting of planning the four years of training for medical residents of a Radiology Department of a hospital in Valencia. The hospital Radiology Department needs to schedule the training of new medical residents each year, establishing a rotating sequence. Throughout the four years of training the residents must perform the following rotations: Technique, Chest, Ultrasound, Digestive, CAT, Muscle-skeletal, Breast, MRI, Vascular, Pediatrics, Nuclear Medicine and Urology. Some rotations are fixed and must be made in a given month. For example, all medical residents must take Technique during the first month of training. Others such as Thorax are essential and should be taken as soon as possible. Some will be taken only after having done other certain rotations. Thus, MRI should be scheduled for residents who have taken the CAT and Ultrasound rotations. Besides the order and duration of the rotations they are characterized by a number of cycles. So, Breast, lasting three months, is done in one cycle of three months, while Gastroenterology, which also lasts three months, has to be completed in two cycles, the first two months during the first year of training and then a further period of one month, to be completed during the third or fourth year. Table 5.13 presents data cycles, durations and conditions to be met in the Radiology Department, regulated by the Specialist Training Guide.

The Radiology Department was interested in optimizing the planning of the training of new medical residents, given the amount of time required and the difficulties involved, they contracted the Polytechnic University of Valencia. This is a combinatorial optimization problem whose objective is to find the best program for training medical residents, taking into account the current legislation and recommendations as to the order, duration and number of times to perform the rotation and the limitations of hospital resources.

This problem was solved with a model with the following definition of variables. R_iXXX_j defines a binary variable that will be worth 1 if the resident i (i = 1, 2, 3) starts the rotating XXX (XXX = 1, 2, ... 12) in month j (j = 1,2, ... 48) and 0 otherwise. For optional rotations other binary variables will be defined (1, 0) R_iOPT_j which means if the doctor does an optional rotation i during the month j (1) or not (0).

Given the combinatorial nature of the problem only the necessary variables should be defined. Binary variables are defined between one starting month and one final month, representing the earliest month to start and the latest month when you can start the rotation. Thus, for Technique which is fixed we will only have one variable for each resident (R_1TEC_1, R_2TEC_1 and R_3TEC_1), while the Chest, Ultrasound, Digestive and CAT rotations will have variables defined between months two and 48. For the remaining seven rotations we will have variables between months eleven and 48, since the first ten months will be occupied by fixed rotations, basic and holidays. The optional rotation will be taken at the end of the training period and therefore we will only define the corresponding variables between months 37 and 48. Logically, we should also define the variables related to the vacation month, which must be in July, August or September. In total, we have 486 variables per resident and therefore 1,458 for the full model, since three new doctors enter each year.

Table 5.13. Cycles, durations and rotation conditions

Rotatory	Cycles and months	Conditions
1. Technical	1	In the first month
2. Chest	2+2+1	Basic
3. Ultrasound	2+2+1	Basic
4. Digestive	2+1	Basic
5. CAT	2+2+1	Basic
6. Muscle-Skeletal	2+2	Independent
7. Breast	3	Independent
8. MRI	2+2	After CAT and ultrasound
9. Vascular	2+2	Independent
10. Pediatrics	3	After CAT and ultrasound
11. Nuclear Medicine	1	Independent
12. Urology	1	Independent
13. Optional	5	4th year

Formulate the objective function and constraints of a model to solve this problem by defining the variables which have been mentioned or using different ones that you may define.

CHAPTER 6

MULTIOBJECTIVE PROGRAMMING AND GOAL PROGRAMMING

All the models for linear and integer programming studied in this book up to now have one feature in common: optimizing a single objective. We have maximized benefits or minimized costs, but always from a single criterion. Selection of this criterion is sometimes difficult when we want to take into account different objectives, which can be in conflict with each other. In this chapter we will present the most commonly used methods to solve problems involving several objectives and goals.

6.1. BASIC CONCEPTS: OBJECTIVES, GOALS AND CRITERIA

In a multiple objective context we first have to define the concept of **attribute**. This concept refers to values related to an objective reality and which can be measured independently of the decision-maker's ideas. Attributes can usually be expressed as mathematical functions of the decision variables. An attribute could be, for example, the benefit. The **objectives** represent addresses to improve the attributes. The improvement can be interpreted as more of a better attribute, corresponding to a maximization process. In the opposite case we will be dealing with a minimization problem. Thus, maximizing profits, maximizing sales, minimizing costs, minimizing risks, etc. are examples of objectives.

We call the acceptable achievement level for the corresponding attribute **aspiration level**. The combination of an attribute with an aspiration level generates a **goal**. Therefore, to obtain a profit of at least 100 million euros is a goal. Summarizing, profit is an attribute, maximizing profit is an objective and reaching a profit at least equal to a given aspiration level is a goal. Finally, **criterion** is a concept that comprises the three previous concepts. In other words, the criteria constitute the attributes, objectives or goals considered as relevant in a decision-making problem.

What is the difference between objectives, goals and constraints? In this particular context the difference between these three concepts is very important. The difference between an objective and a goal is clear. While the goal establishes an acceptable aspiration level of the attribute, the objective optimizes it, maximizing or minimizing. The difference between a goal and a constraint is more subtle. In mathematical terms, a goal looks exactly the same as a constraint. However, goals involve more flexibility and less stiffness than constraints. For a goal the second member is simply a value we aspire to. However, in a constraint we must always fulfil the right-hand side, because otherwise the constraint will be violated and we would have a non-feasible solution. In single-objective mathematical programming the concept of goal does not exist, working only with constraints.

Another important concept is the **trade-off rate between criteria**. The trade-off rate between two criteria means the amount of achievement of a criterion that must be sacrificed to obtain a unitary increase in another criterion. The advantages of trade-off rates are twofold in the multiple-objective approach. On the one hand, they are a good index for measuring the opportunity cost of a criterion in terms of the other criteria we are also considering. On the other hand, the concept of trade-off rate plays an important role in the development of interactive multiple-objective methods. The interaction becomes a kind of dialogue in which the decision-maker transmits his preferences measured by the trade-off rates to the system analyst.

The conceptual difference between attributes, objectives and goals allows us to make a first methodological classification of the different multiple criteria approaches. Therefore, when decision making lies within a context with multiple objectives we can apply multiobjective programming. If the process of decision making takes place within a context with multiple goals, we will then consider goal programming. Both approaches, multiobjective and goal programming, can solve continuous and discrete problems. Nevertheless, there are multiple criteria methods especially designed for problems with a discrete number of feasible solutions, such as AHP and PROMETHEE that will be explained in chapter 7.

6.2. MULTIOBJECTIVE PROGRAMMING

There are many situations in which it is necessary to consider different characteristics that are difficult to combine in a single objective: investment profitability and risk, short-term benefits and company long-term growth, services costs and quality, etc. Usually, multiobjective problems can have intrinsically different or intrinsically similar objectives. In the first case the opportunity cost of an objective over the others can sometimes be determined and in other cases the criteria can be ordered by importance. Multiobjective programming and goal programming are useful techniques for solving problems in this context.

One of the many real problems in which multiobjective programming is applied is natural resources management. For example, the administration can decide to include profitability, historical, ecological, environmental, hydraulic values, etc. in land management and planning. In many occasions the aim is to preserve public lands in their natural conditions to provide shelter and food for wild life and domestic animals while also considering the recreational use of the land. Many such objectives conflict with each other. Farming associations want more land for farming and/or pasture, mining companies want exploitation rights and groups for the preservation of the environment want to maintain some areas in their natural state.

Another application is the construction of airports in large cities. Possible objectives could be: minimizing the overall costs of construction, maintenance and servicing, minimizing the access time to the airport, maximizing safe operations, minimizing the effects of noise pollution caused by air traffic and increasing the airport's air traffic capacity.

We will see the main concepts of multiobjective programming with a simple example. This is the problem: a plot of land that is currently devoted to hunting wants to be partially transformed to be used for cattle breeding with the objective of increasing benefits without totally renouncing its recreational value and trying to maintain in both cases a level of exploitation that guarantees the ecological balance.

The decision variables of the model, X_1 and X_2, represent the hectares devoted to sheep pasture and those left for hunting activities, respectively. The two objectives to maximize are profit and recreational value of the plot of land. The function representing the latter has been built from an arbitrary relative scale from 0 to 4, giving the maximum value when the land is only exploited for hunting, and value 1 when the cattle is introduced. For different reasons, it is considered that no more than 1500 ha should be devoted to pasture, and that no more than 1000 ha should be exclusively devoted to hunting. The total surface area of the plot of land is 2,000 ha.

Max[Benefit $Z_1 = 30 X_1 + 5 X_2$; Recreational Value $Z_2 = X_1 + 4X_2$]

Land surface: $X_1 + X_2 \leq 2000$

$$X_1 \leq 1500$$
$$X_2 \leq 1000$$
$$X_1, X_2 \geq 0$$

Figure 6.1 represents the feasible region and the optimal solution if maximizing only one single objective. If the objective is profit, the optimal solution will be point B and if the objective is to maximize the recreational value, the optimal solution will be point C. With the optimal values of both objective functions we will build the pay-off matrix of Table 6.1. Thus, when maximizing benefit, it gives a value of 47,500 monetary units and a recreational value of 3,500, while when maximizing the recreational value, its value will be 5,000 and the benefit will be lower than in the former case, more specifically 35,000 monetary units.

The point (Z_1=47,500, Z_2=5,000) formed by the elements of the pay-off matrix diagonal is called **ideal point**, where the two objectives reach their optimal value and this is usually impossible. This point and the **anti-ideal point**, which is the point at which Z_1 is 35,000 and Z_2 is 3,500, are important in the solving methods of multiobjective problems. The anti-ideal point is a "bad solution", but it is useful, for example, to "standardize" the objectives measured in the different units and with different absolute values. In addition, the difference between ideal and anti-ideal values defines a range of values for each criterion that it is necessary to know when using certain resolution methods for multiobjective problems.

Table 6.1 Pay-off matrix

Maximizing	Benefit	Recreational Value
Benefit	47,500	3,500
Recreational value	35,000	5,000

Table 6.2 shows the coordinates of the corner-points of OABCD in the decision variable space of Figure 6.1, and their corresponding objectives graphically represented in Figure 6.2 by points O', A', B', C' and D'.

Table 6.2. Values of decision variables and objectives at corner-points

Corner-point	Decision variables		Objectives	
	X1 Ha Pasture	X2 Ha hunting	Z1 Benefit (thousands mu)	Z2 Recreational Value
O	0	0	0	0
A	1,500	0	45,000	1,500
B	1,500	500	47,500	3,500
C	1,000	1,000	35,000	5,000
D	0	1,000	5,000	4,000

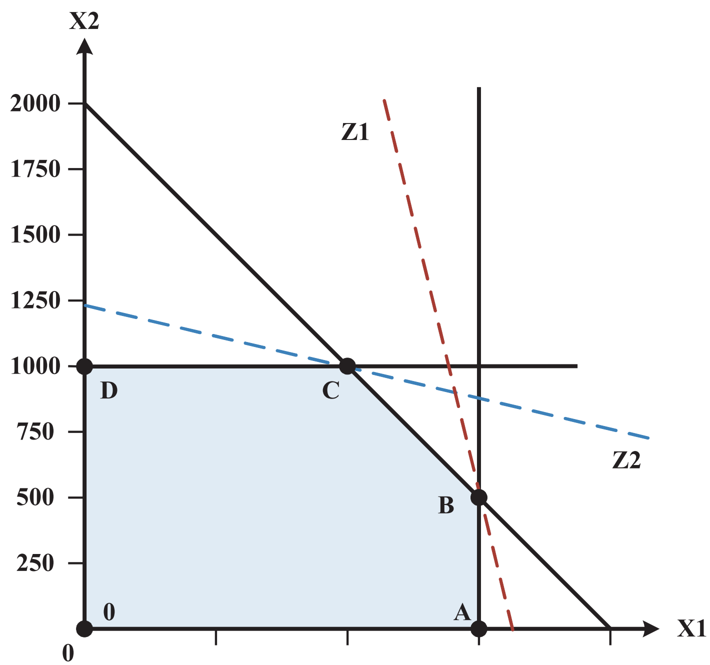


Figure 6.1 Feasible region and optimal solution with single objective

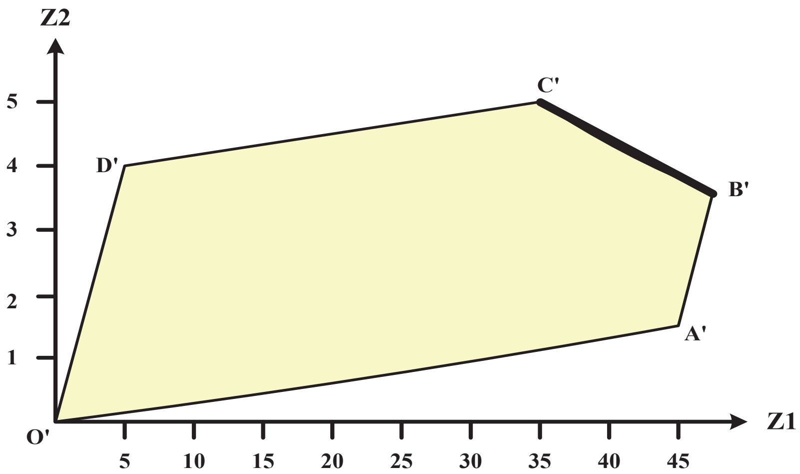


Figure 6.2. Efficient set in the objective space

When there are objectives in conflict, as in this case, the ideal point where all the objectives reach their optimal solution does not exist. Therefore, what the resolution of a multiobjective model does is to find the efficient set. A point belongs to the **efficient set** or **boundary** when it fulfils the constraints and there is no other possible solution to improve one of the objectives without worsening at least another one. As shown in Figure 6.2, it is evident that the efficient boundary in our problem is segment C'B'. Note that the efficient set can be defined in terms of efficient corner-points (B´ and C´) or in terms of corner and interior points (segment B´C´). It is also worth noting that the slope of segment B´C´ provides the trade-off rate between objectives and is therefore a measurement of the opportunity cost of one for the other.

With the set of efficient points we can select one solution, regarding the subjective criteria in small problems. On the other hand, we have been able to graphically solve the previous problem because it is a model with two decision variables and two objectives. Larger problems can be solved using different methods. Among them we find the constraints method and the weight method.

6.2.1. CONSTRAINTS METHOD

This method consists of optimizing one of the objectives, while the other objectives are treated as parametric constraints. It has been proved that for each set of values of the right-hand sides an efficient point is obtained, either corner or interior point. Therefore, to solve the problem linear programming software can be used. The model would be the following

$$MAX\ Z_1 = 30\ X_1 + 5\ X_2$$

$$X_1 + X_2 \leq 2{,}000$$

$$X_1 \leq 1{,}500$$

$$X_2 \leq 1{,}000$$

$X_1 + 4\ X_2 \geq P$ the value of P should be between 3,500 y 5,000

For each value of parameter P an efficient point is generated. The range of P will be defined by both values, the ideal and anti-ideal points. Table 6.3 shows some points of the efficient boundary. It is important to know that this method provides efficient solutions only when the parametric constraints are verified strictly in the optimal solution. This happens in all the points presented in Table 6.3 because of that Z_2 and P coincide.

Table 6.3 Efficient points generated from the constraints method

POINT	X1	X2	Z1	Z2	P
B	1,500	500	47,500	3,500	3,500
1	1,416.6	583.3	45,416.6	3,750	3,750
2	1,333.3	666.6	43,333.3	4,000	4,000
3	1,250	750	41,250	4,250	4,250
4	1,166.6	833.3	39,166.6	4,500	4,500
5	1,083.3	916.6	37,083.3	4,750	4,750
C	1,000	1,000	35,000	5,000	5,000

Generally, the constraints method for n objectives will be

$$MAXIMIZING\ Z_k\ (X)$$

$$Z_j\ (X) \geq P_j \quad for\ j = 1,2,..., k\text{-}1, k+1,..., n$$

Constraints of the model

In the objectives to minimize that appear as parametric constraints, the direction of the constraint will be reversed. Thus, if we had **to minimize Z_1** we would set

$$\mathbf{Z_1(X) \leq P_1}$$

6.2.2. WEIGHT METHOD

This technique consists of aggregating all the objectives in a function where a nonnegative weight W is assigned to each objective in the following way

$$MAX\ W_1\ Z_1\ (X) + W_2\ Z_2\ (X) + ... + W_n\ Z_n(X)$$

Constraints of the model

Where $W_i \geq 0$ and the constraints of the problem are fulfilled. Parametrically when changing weights W the efficient set can be generated or at least approximated. With this method only efficient-corner points are generated, contrary to the previous method in which corner and interior points are generated. Why? Think about sensitivity analysis of right-hand sides of constraints as well as sensitivity analysis of objective function coefficients.

In our example, the model will be the following:

$$\mathbf{MAX} \quad \mathbf{W_1\ (30\ X_1 + 5\ X_2) + W_2\ (X_1 + 4\ X_2)}$$

$$X_1 + X_2 \leq 2{,}000$$

$$X_1 \leq 1{,}500$$

$$X_2 \leq 1{,}000$$

$$\mathbf{Z_1 = 30\ X_1 + 5\ X_2}$$

$$\mathbf{Z_2 = X_1 + 4\ X_2}$$

For example, for values of $W_1 = 1$ and $W_2 = 2$ efficient corner-point B’ is obtained and for $W_1 = 1$ and $W_2 = 10$ point C’ is obtained (Figure 6.2).

It is necessary to emphasize that this method, as well as the other multiobjective methods, do not take into account the decision-maker's preferences. Therefore, the W_i chosen do not have a relationship with those preferences. These weights act as arbitrarily changed parameters in order to generate the efficient corner-points of the problem under consideration. Finally, both the constraints method and the weight method only guarantee approximations to the efficient set. No matter how detailed the parametric analysis which has been performed, we can never be sure that some efficient points have been omitted.

6.2.3. OTHER MULTIOBJECTIVE TECHNIQUES

Another method is the NISE (Noninferior set estimation) multiobjective method. This method consists of the iterative application of the weight method, though in this case the weights are not chosen arbitrarily, but the quotient of their values (W_1/W_2) is equal to the slope of the line that joins the efficient points obtained in the previous iteration. With this method we can obtain a fast and good approximation to the efficient set. Even for medium-size problems it can generate an exact representation of the efficient set.

The only method that guarantees the generation of all efficient corner-points is the simplex multiobjective method. The purpose of this method, first proposed by Philip (1972) and Zeleny (1973), consists of finding all efficient corner-points of a multiobjective problem, moving from a corner-point to an adjacent corner-point. The conventional simplex algorithm constitutes the mechanism suitable for performing this type of search.

Finally, there is an operational problem in all multiobjective methods which consists of the high number of efficient points generated. This problem of excessive information can be solved by decreasing the efficient point set in a number of ways, among which we find pruning and filtering techniques. With these techniques efficient solutions which are not very different from other solutions previously obtained are rejected. The generation of efficient solutions in a multiobjective problem can become difficult when the number of objectives and constraints increases. Even if found, the number of solutions may be so high that the selection of the optimal solution may become a difficult task. In these cases, other more pragmatic multiple criteria approaches such as goal programming models can be used instead.

6.3 GOAL PROGRAMMING

6.3.1. GENERAL STRUCTURE OF A GOAL PROGRAMMING MODEL

Goal programming is based on quantitatively establishing an acceptable achievement level for each of the objectives and then search for the solution that minimizes the weighted sum of the deviations for each objective over a fixed numerical value. Goal programming follows the philosophy proposed by Herbert Simon (Nobel in Economy, 1978). According to Simon present decisional context is defined by incomplete information, limited resources, multiple objectives, interest conflicts, etc. In this complex context, the decision maker tries to approach a set of relevant goals as much as possible to some previously fixed aspiration levels.

In section 6.1 the concepts of attribute, objective, aspiration level and goal have been defined. The first step in the formulation of a goal programming model consists of setting those attributes considered relevant for the problem under analysis. The next step is to determine the aspiration level for each attribute, i.e. the achievement level we wish to reach.

Thus, for the i-th attribute, the goal will be as follows

$$f_i\,(x) + N_i - P_i = t_i$$

Where $f_i\,(x)$ is the mathematical expression of the i-th attribute, t_i its aspiration level, N_i and P_i are the negative and positive deviation variables, respectively.

The negative deviation variables quantify the lack of achievement of a goal with respect to its aspiration level, whereas the positive deviation variables quantify the surplus of achievement of a goal with respect to its aspiration level. Table 6.4 presents the three types of goal that we can formulate. The first case is an upper one-sided goal, the second is a lower one-sided goal and the third a two-sided goal.

A goal is not the same as a constraint, although mathematically they are expressed in a similar way. A constraint must be always fulfilled; otherwise the model will not have any feasible solutions. However, the goal represents the level that we want to reach and we can therefore stand above or below that level. Depending on the type of goal, we will have to choose the deviation variables to be minimized. For example, if the goal is to reach a minimum profit level of 20 million euros, the deviation variable to be minimized will be the negative deviation, if we reach more than that level the better. On the other hand, if we want our investment to be at most 8 million euros, the variable to be minimized in this case, which is an upper one-sided goal, will be the positive deviation. We will minimize only the sum of the two deviation variables when we want to reach a given level exactly, as in the case of wanting to maintain the work force of the company.

Since a goal cannot simultaneously exceed or be below the aspiration level at least one of the two deviation variables must be zero. Both deviation variables will take a zero value when the goal strictly reaches its aspiration level.

Table 6.4 Formulation of goals and deviation variables

Type of goal	Formulation of goal	Deviation variables to minimize
$f_i(x) \leq t_i$	$f_i(x) + N_i - P_i = t_i$	P_i
$f_i(x) \geq t_i$	$f_i(x) + N_i - P_i = t_i$	N_i
$f_i(x) = t_i$	$f_i(x) + N_i - P_i = t_i$	$N_i + P_i$

We will learn the main concepts of goal programming using a simple example of production planning (Ballestero and Romero, 1998). A public paper company manufactures two types of products, cellulose pulp obtained by mechanical systems, and cellulose pulp obtained by chemical processing. The maximum production capacities are estimated at 300 and 200 ton/day for each type of cellulose pulp. One worker is required to produce one ton of cellulose pulp. The company has of a work force of 400 workers, and does not want to take on more personnel.

The gross margin (income minus variable costs) per ton of cellulose paste obtained mechanically is estimated at 1,000 m.u. and that obtained chemically at 3,000 m.u. The fixed costs of the paper company are estimated at 400,000 m.u./day. The company would like, to cover fixed costs at least. The preferences of the company are the maximization of the gross margin (economic objective) and the minimization of the damage to the river which the paper company spills its wastes into (environmental objective). It is estimated that the production waste generated per ton of mechanically and chemically obtained cellulose paste generate biological demands for oxygen in the river water of 1 and 2 units, respectively.

Let's consider an aspiration level of 300 units for the biological demand of oxygen. The right-hand side of the corresponding constraint will be the aspiration level associated with the attribute. We will suppose a value of 400,000 m.u. for the gross margin.

Thus, the company goals are the following:

Biological demand for oxygen: $X_1 + 2X_2 + N_1 - P_1 = 300$
Gross margin: $1000X_1 + 3000X_2 + N_2 - P_2 = 400{,}000$
Employment: $X_1 + X_2 + N_3 - P_3 = 400$
Production capacity X_1: $X_1 + N_4 - P_4 = 300$
Production capacity X_2: $X_2 + N_5 - P_5 = 200$

Let's see what the undesirable deviation variables are. For the first goal, the undesirable variable would be P_1, since the lower the biological demand for oxygen better. For the second goal, gross margin, the variable to be minimized will be N_2 since, it is better if we reach a gross margin higher than the established one. For the goal of employment, for which the idea is to maintain current levels, we have to minimize both the positive and the negative variables. Finally, we do not wish to exceed the production capacities, which would involve working extra shifts, therefore the undesirable variables are P_4 and P_5.

6.3.2. WEIGHTED GOAL PROGRAMMING

The goal programming model is solved by minimizing the sum of the undesirable deviation variables. In our example we would have

$$MIN\ P_1 + N_2 + N_3 + P_3 + P_4 + P_5 \quad (1)$$

This expression is the sum of the variables measured in different units, which is meaningless. In addition, the absolute values of the aspiration levels are highly different. Therefore, we could have solutions shifted towards the goals with high aspiration levels. We can avoid these problems, if instead of minimizing the sum of absolute deviations we minimize the sum of the deviations measured as a percentage. Thus, (1) becomes

$$MIN \quad \frac{P_1}{300} + \frac{N2}{400000} + \frac{N3 + P3}{400} + \frac{P4}{300} + \frac{P5}{200} \quad (2)$$

As percentages are dimensionless, the sum in the previous expression does not present any problems of homogeneity. Furthermore, the standardization procedure used removes any bias to the fulfilment of goals with high aspiration levels. However, in expression (2) it is assumed that we place the same importance on all of the goals, which is not necessarily always the case. This problem can be solved by replacing expression (2) by

$$MIN \quad W_1 \frac{P_1}{300} + W_2 \frac{N_2}{400000} + W_3 \frac{N_3 + P_3}{400} + W_4 \frac{P_4}{300} + W_5 \frac{P_5}{200} \quad (3)$$

Where the W_i coefficients are the relative importance we wish to give to each goal. This method consists of minimizing the weighted sum of the undesirable deviation variables, expressed in percentage; this is known as weighted goal programming. In our example, the overall formulation of the weighted goal model is the following:

$$MIN \quad W_1 \frac{P_1}{300} + W_2 \frac{N_2}{400000} + W_3 \frac{N_3 + P_3}{400} + W_4 \frac{P_4}{300} + W_5 \frac{P_5}{200} \quad (3)$$

Subject to

Biological demand for oxygen: $X_1 + 2X_2 + N_1 - P_1 = 300$

Gross margin: $1{,}000X_1 + 3{,}000X_2 + N_2 - P_2 = 400{,}000$

Employment: $X_1 + X_2 + N_3 - P_3 = 400$

Production capacity X_1: $X_1 + N_4 - P_4 = 300$

Production capacity X_2: $X_2 + N_5 - P_5 = 200$

This is a linear programming model and thus we can solve it through the simplex algorithm. For different weights different solutions will be generated. If we give the same importance to all goals, all W_i will equal 1. The solution obtained through LINGO is the following.

```
MODEL:

! Production planning

Xi= Ton/day, cellulose pulp obtained by mechanical systems (1) and
    chemical processing (2);

[OF] MIN = 33.33*P1 + 0.025*N2 + 25*N3 + 25*P3 + 33.33*P4 + 50*P5;

! Constraints: GOALS;

[GOAL_BDO] X1 + 2*X2 + N1 = 300 + P1;

[GOAL_GMARGIN] 1000*X1 + 3000*X2 + N2 = 400000 + P2;

[GOAL_ EMPLOYMENT] X1 + X2 + N3 = 400 + P3;

[GOAL_PCX1] X1 + N4 = 300 + P4;

[GOAL_PCX2] X2 + N5 = 200 + P5;

[BDO] X1 + 2*X2 = VBDO;

[GM] 1000*X1 + 3000*X2 = VGM;

[EMPLOYMENT] X1 + X2 = VEMPLOYMENT;

END
```

```
Global optimal solution found.
  Objective value:                    3888.66

  VARIABLE            VALUE            REDUCED COST
        P1         66.666664              0.000000
        N3         66.666664              0.000000
        X1        300.000000              0.000000
        X2         33.333332              0.000000
        N5        166.666672              0.000000
      VBDO        366.666656              0.000000
       VGM     400000.000000              0.000000
  EMPLOYMENT      333.333344              0.000000
```

In this solution we can see that we get the gross margin established as goal (400,000 m.u.) exactly, exceed the goal of biological demand for oxygen by 66.6 and obtain a level of employment lower than the goal. Similarly, the production of cellulose paste obtained by mechanical systems completely reaches the goal established in terms of production capacity, whereas the production of cellulose paste obtained by chemical processes is very small. In the model solved with LINGO the mathematical expressions for the biological demand for oxygen, the gross margin and employment have been included to see the levels reached in the optimal solution. Note that the right-hand side of these three expressions are VBDO, VGM and VEMPLOYMENT, variables for whose only purpose is to show the value of biological demand for oxygen, gross margin and the employment level in the optimal solution. Variables which are not shown in the optimal solution are zero. Analysis based on goal programming can be improved by using sensitivity analysis for the weighted coefficients.

6.3.3 PREEMPTIVE GOAL PROGRAMMING

In weighted goal programming all goals are supposed to have comparable importance. However, there are situations in which some goals have priority over other goals; in this case we have **preemptive goal programming or lexicographic goal programming**, and the goals are first classified into first-priority, second-priority, third priority, etc. The goals with the highest priority are satisfied as much as possible, only after this the possible satisfaction of goals with lower priorities will be considered. In other words, preferences are ordered in the same way as the words in a dictionary which is why this type of programming is also referred to as lexicographic goal programming.

In order to illustrate this approach we are going to modify the example used in the previous section. Supposing that first priority Q_1 consists of the goals of the factory's production capacities. The second priority Q_2 consists of the goal that states that the biological demand for oxygen be at most 300 units. Priority Q_3 is gross margin and the forth priority Q_4 is the employment goal. Therefore, the overall process of lexicographic minimization of the undesirable deviation variables is given by the following vector:

$$Lex\ min\ a = [(P_4 + P_5), (P_1), (N_2), (N_3 + P_3)]$$

This vector is known as an **achievement function** and replaces the conventional objective function. Each component of the achievement function represents the deviation variables to be minimized, with the purpose of obtaining the maximum possible achievement of the goals in the corresponding priority. Generally, the achievement function is represented as

$$Lex\ min\ a = [h_1 (N, P), h_2 (N, P),\dots,h_k (N,P)]$$

Or in abbreviated form

$$Lex\ min\ a = [a_1, a_2\dots a_k]$$

Where a_k = h (N, P) is a function of the undesirable deviation variables. The lexicographic minimization of the previous vector involves the ordered minimization of its components. That is, first the smallest value of component a_1 is found, then the smallest value of component a_2, compatible with the value of a_1, and so on.

The preemptive goal programming model for our example will be:

Lex min $a = [(P_4 + P_5), (P_1), (N_2), (N_3 + P_3)]$

Q_2: Biological demand for oxygen: $X_1 + 2\ X_2 + N_1 - P_1 = 300$

Q_3: Gross margin: $1{,}000\ X_1 + 3{,}000\ X_2 + N_2 - P_2 = 400{,}000$

Q_4: Employment: $X_1 + X_2 + N_3 - P_3 = 400$

Q_1: Production capacity X_1: $X_1 + N_4 - P_4 = 300$

Production capacity X_2: $X_2 + N_5 - P_5 = 200$

This preemptive goal programming model can be solved with algorithms based on the simplex algorithm. The fundamental difference between linear programming and preemptive goal programming is that in the first we seek one point (corner-point) that maximizes a single objective, whereas in goal programming we seek a region that provides a commitment between a set of conflicting goals.

We solve this preemptive goal programming model using a sequence of linear programming models. Firstly, we solve the linear programming model with the first level goals.

```
MODEL:
! Chapter 6. Preemptive goal programming;
Production planning
Xi= Ton/day, cellulose pulp obtained by mechanical systems (1) and
chemical processing (2);
[OF_Q1] MIN = P4 + P5;
! Constraints: GOALS;
[GOAL_BDO] X1 + 2*X2 + N1 = 300 + P1;
[GOAL_GMARGIN] 1000*X1 + 3000*X2 + N2 = 400000 + P2;
[GOAL_ EMPLOYMENT] X1 + X2 + N3 = 400 + P3;
[GOAL_PCX1] X1 + N4 = 300 + P4;
[GOAL_PCX2] X2 + N5 = 200 + P5;
[BDO] X1 + 2*X2 = VBDO;
[GM] 1000*X1 + 3000*X2 = VGM;
[EMPLOYMENT] X1 + X2 = VEMPLOYMENT;
 END
```

The deviation variables P_4 and P_5 are zero, so we set these values into the model to minimize the deviation variables of the second-priority goals.

```
MODEL:
!Chapter 6. Preemptive goal programming;
Production planning
Xi= Ton/day, cellulose pulp obtained by mechanical systems (1) and
chemical processing (2);
[OF_Q2] MIN = P1;
! Constraints: GOALS;
[GOAL_BDO] X1 + 2*X2 + N1 = 300 + P1;
[GOAL_GMARGIN] 1000*X1 + 3000*X2 + N2 = 400000 + P2;
[GOAL_ EMPLOYMENT] X1 + X2 + N3 = 400 + P3;
[GOAL_PCX1] X1 + N4 = 300;
[GOAL_PCX2] X2 + N5 = 200;
[BDO] X1 + 2*X2 = VBDO;
[GM] 1000*X1 + 3000*X2 = VGM;
[EMPLOYMENT] X1 + X2 = VEMPLOYMENT;
END
```

We obtain an optimal solution with $P_1= 0$ by solving the model for the second-priority goal. That means that we achieve the goal of biological demand for oxygen. The next step is to solve a model by minimizing the negative deviation of the gross margin goal (N_2), being $P_1=0$ and we obtain $N_2=0$ in the optimal solution. Note that the latter variable does not appear in the next linear programming model, because $N_2=0$.

```
MODEL:
! Chapter 6. Preemptive goal programming;
Production planning
Xi= Ton/day, cellulose pulp obtained by mechanical systems (1) and
chemical processing (2);
[OF_Q3] MIN = N2;
! Constraints: GOALS;
[GOAL_BDO] X1 + 2*X2 + N1 = 300;
[GOAL_GMARGIN] 1000*X1 + 3000*X2 + N2 = 400000 + P2;
[GOAL_ EMPLOYMENT] X1 + X2 + N3 = 400 + P3;
[GOAL_PCX1] X1 + N4 = 300;
[GOAL_PCX2] X2 + N5 = 200;
[BDO] X1 + 2*X2 = VBDO;
[GM] 1000*X1 + 3000*X2 = VGM;
[EMPLOYMENT] X1 + X2 = VEMPLOYMENT;
END
```

Finally, we solve the linear programming model that minimizes the deviation variables of employment goals. The objective function is the summation of N_3 and P_3 because we prefer to maintain the employment level, without increasing or decreasing the number of workers.

```
MODEL:
! Chapter 6. Preemptive goal programming;
Production planning
Xi= Ton/day, cellulose pulp obtained by mechanical systems (1) and
chemical processing (2);
[OF_Q4] MIN = N3 + P3;
! Constraints: GOALS;
[GOAL_BDO] X1 + 2*X2 + N1 = 300;
[GOAL_GMARGIN] 1000*X1 + 3000*X2 = 400000 + P2;
[GOAL_ EMPLOYMENT] X1 + X2 + N3 = 400 + P3;
[GOAL_PCX1] X1 + N4 = 300;
[GOAL_PCX2] X2 + N5 = 200;
[BDO] X1 + 2*X2 = VBDO;
[GM] 1000*X1 + 3000*X2 = VGM;
[EMPLOYMENT] X1 + X2 = VEMPLOYMENT;
END
```

Global optimal solution found.

Objective value: 200.0000

Variable	Value	Reduced Cost
N3	200.00	0.000000
X1	**100.00**	0.000000
X2	**100.00**	0.000000
N4	200.00	0.000000
N5	100.00	0.000000
VDBO	**300**.00	0.000000
VMARGEN	**400000**.00	0.000000
VEMPLEO	**200**.00	0.000000

We see that the optimal achievement vector is a = [0, 0, 0, 200]. Therefore, the solution allows the total achievement of the first-, second- and third-priority goals, whereas for the fourth priority, there is a negative deviation of 200 units.

Although variables P_4 and P_5 are measured in the same units (ton/day) and their sum is meaningful, the aspiration levels are different, and therefore the term $P_4 + P_5$ of the achievement function should be replaced by $(P_4/300) + (P_5/200)$ as indicated in the previous section. It is also useful to compare the results obtained in the weighted goal model and in the preemptive goal model. Analyse their differences. The results obtained with goal programming models can be improved by analysing the influence of priority hierarchy on the solution.

6.4. SUMMARY

Firstly, we defined the basic concepts of the multiple criteria approaches: attributes, objectives and goals. Secondly, we have illustrated the multiobjective programming approach with a simple example, consisting of two decision variables and two objectives, to be solved graphically. This allowed us to understand the concept of **efficient solution or Pareto optimum**, which is the solution in which we cannot improve one objective without worsening at least another objective. Multiobjective programming establishes the set of efficient solutions. Although this approach is different from conventional mathematical programming for optimizing a single objective, operational practice is similar in the sense that it consists of solving models using the well-known simplex algorithm if we have a linear programming model. The efficient set of solutions is approximated or generated optimizing one of the objectives and parameterizing the others. Nevertheless, when the number of objectives and constraints increases, the decision making becomes more complex. In these cases goal programming could be a more pragmatic approach. Thus, in this chapter we have explained the general structure of a goal programming model as well as weighted and preemptive goal programming models. The fundamental difference between linear programming and preemptive goal programming is that in the first we seek one point (corner-point) that maximizes a single objective, whereas in goal programming we seek a region that provides a commitment between a set of conflicting goals.

6.5. SELECTED REFERENCES

1. Ballestero, E. and Romero, C. (1998): Multiple Criteria Decision Making and its Applications to Economic Problems. Kluwer Academic Publishers.

2. Daellenbach, H.G.; George,J.A. y McNickle,D.C. (1987): Introducción a las técnicas de Investigación de Operaciones.

3. Eppen, G.D.; Gould, F.J.; Schmidt, C.P.; Moore, J.H. and L.R. Weatherford (1997): Introductory Management Science: Decision Modeling with Spreadsheets. 5th edition. Pearson.

4. LINDO Systems (2011): LINGO. The modeling language and optimizer.

5. Zopounidis, C. and Pardalos P.M. (Eds.) (2010): Handbook of Multicriteria Analisis. Springer.

6. Winston, W. L. and S.C. Albright (2012): Practical Management Science. Fourth Edition. South-Western. USA.

6.6. CASE STUDIES

CASE STUDY 1: PRODUCTION PLANNING PROBLEM

A public paper company manufactures two types of products, cellulose pulp obtained by mechanical systems, and cellulose pulp obtained by chemical processing. The maximum production capacities are estimated in 300 and 200 ton/day for each type of cellulose paste. Each ton of cellulose paste requires one worker. The company has a work force of 400 workers, and does not want to take on more workforces.

The gross margin (income minus variable costs) per ton of cellulose paste obtained mechanically is estimated at 1,000 m.u. and that obtained chemically at 3,000 m.u. The fixed costs of the paper company are estimated at 300,000 m.u. /day. The company would like, to cover fixed costs at least.

The preferences of the company are to maximize the gross margin (economic objective) and minimize the damage to the river into which the paper company pumps its production waste (environmental objective). It is estimated that the production waste per ton of mechanically and chemically obtained cellulose paste generates biological demands for oxygen in the river water of 1 and 2 units, respectively.

1. Formulate a model that allows the company to decide its production, taking into account the objective of maximizing the gross margin, and of minimizing the environmental damage caused.
2. Graphically solve the previous model, representing the feasible set in the decision variable space and the efficient set in the objective space.
3. Calculate the trade-off rate between objectives.
4. Calculate the pay-off matrix.
5. Generate at least five efficient points using the constraints method.
6. Generate two efficient points with the weight method.

CASE *STUDY 2: MULTIPLE LAND USE MANAGEMENT*

A consulting company has to write a report on the best decisions that can be taken by the management of a natural park. There are several production values that are desired. Two of the productive activities that can be carried out are coal mining and sheep herding. Two recreational activities that can be considered for the park are hunting and track vehicle usage. The board of directors consider that adequate forest management is needed in order to avoid erosion (desertification), to facilitate the absorption of water and CO_2, and to improve the recreational and landscaping value of the natural park.

The natural park is divided into two zones; A and B. Zone A, which is 1,200 hectares, can be dedicated to coal mining, sheep herding, hunting and forestry. Zone B (5,400 hectares) can be dedicated to sheep herding, hunting, vehicle and forestry.

Some activities are not mutually compatible and cannot be carried out on the same land surface. For example, forestry is not compatible with mining and sheep herding. In other words, these three activities are mutually and collectively exclusive. Furthermore, coal mining is also incompatible with hunting and sheep herding. Lastly, sheep herding and hunting are compatible, but sheep production is reduced if hunting is allowed in the same area. Similarly, the recreational value of an area dedicated to hunting and sheep herding is less than when there is only hunting in the area. As a result of all this we differentiate the following simultaneous activities over areas: 1) Sheep herding, 2) Hunting, 3) Sheep herding and hunting. All the data is collected in the following table.

1. Formulate a multiobjective linear programming model that would allow the consulting firm to complete a report.

2. Calculate the complete pay-off matrix with the five objectives.

3. Calculate at least two efficient points. State the solution and the values of all objectives for the two efficient points.

4. Propose a management plan for the natural park based on the previous analysis.

Table 6.5 Productions for each type of land used per hectare

Production	Sheep herding	Coal mining	Sheep herding and hunting	Hunting	Forestry	Vehicle
Animals/year	3	0	2.5	2	0	0
Coal (thousands of tons/year)	0	4	0	0	0	0
Recreational value	0	-5	1	0	0	3
Erosion avoidance	-1	-3	0	1	5	-3
NPV (m.u.)	300	1,000	320	20	-50	-40

CASE STUDY 3

A company is considering the production of three new products to replace the current models and wants to know how many of the new products it should manufacture. The management wants to give primary consideration to three factors: Net Present Value, work force stability and capital investment required for the new equipment.

The established goals are:

1. Achieving a NPV of at least 125 million euros.

2. Maintaining current employment level of 4,000 employees

3. Holding the capital investment to less than 55 million euros.

However, the management realizes that it will probably not be possible to attain all of these three goals simultaneously, so they have discussed their priorities. This discussion has led to setting penalty weights of 15 for missing the NPV goal (per million euros below), 2 for going over the employment goal (per hundred workers), 4 for going under the same goal and 5 for exceeding the capital investment goal (per million euros above).

The contribution of each new product to NPV, employment level and capital investment level are proportional to the rate of production. Table 6.6 shows the contributions per production unit, and the goals and penalties.

1. Formulate and solve with LINGO or Excel the weighted goal programming model that allows the company to decide the quantity of new products to manufacture, as well as the achievement level for each goal.

2. As the optimal solution obtained has not convinced the company, the management has decided that a very high priority should be placed on avoiding an increase in the work force. Furthermore, the management knows that raising capital of more than 55 million euros to invest in the new products would be extremely difficult, so a very high priority should also be placed on avoiding capital investment above this level.

Based on these considerations, the management has concluded that a preemptive goal programming approach should be used, where the two goals, just discussed, should be the first-priority goals, and the other two original goals (exceeding 125 million euros in NPV and avoiding a decrease in the employment level) should be the second-priority goals. The relative penalty weights should still be the same as given in Table 6.6.

Formulate the corresponding model and obtain the optimal solution with LINGO or Excel, analysing the results obtained.

Table 6.6. Attributes, product contribution and goals

Attribute	Unit Contribution			Goal	Penalty weight
	product 1	product 2	product 3		
PNV	12	9	15	≥ 125	15
Employment	5	3	4	= 40	2(+), 4(-)
Investment	5	7	8	≤ 55	5

Table 6.7. Revised formulation: preemptive goal programming

Priority level	**Attribute**	**Goal**	**Penalty Weight**
First priority	Employment Investment	≤ 40 ≤ 55	2 5
Second Priority	NPV Employment	≥125 ≥40	15 4

CASE STUDY 4: SELECTION OF PUBLICITY MEDIA

A company manufactures and distributes a seasonal product in a region that comprises a large metropolitan area. The product is especially attractive for families with children. The company has offered a discount promotion on the product before the season and is planning a marketing campaign. It is considering an advertisement with full page colour ads in the supplements of the Sunday editions of the two most important newspapers. The slogan and copy have been prepared. The only aspect to determine is the time planning of the mass media; that is, the number of consecutive inserts in each newspaper.

From an ideal point of view, this should be related to the profit obtained for each insert. However, it is very difficult to measure the profit of time planning in a mass media. Therefore, in practice, alternative measurements are used which have proved to be positively related with profit. Examples of these measurements include scope of the time programming of the mass media (defined as the fraction of people in a given population of clients exposed to the advertisements at least once) and the time planning frequency (defined as the average number of exposures among the members of the population who have paid attention to the advertisement at least once). It would also be convenient to reach different sectors of the population in a different extent. Let us consider that for this problem scope is the most suitable criterion and that we want to distinguish between a primary group formed by all of the people having at least one elementary schoolchild (goal 1) and a secondary group that covers all the families with annual income higher than 8,000 euros (goal 2).

From the newspapers some data can be obtained about the scopes of the Sunday newspaper supplements, corresponding to the different population groups. For example, for newspaper X and the primary group, they indicate the following average fraction of people reached in the group as a function of the number of inserts:

Table 6.8 Scope of time programming for newspaper X (primary group)

Number of inserts x	1	2	3	4	5	6
Accumulated fraction y	0,54	0,66	0,75	0,83	0,87	0,89

Unfortunately, this is a nonlinear relationship with a significant drop of accumulated scope after 4 inserts. The same holds for the secondary group in newspaper X. Therefore, it seems unprofitable to exceed four inserts in newspaper X. During the first four inserts it is possible to approach the previous relationship closely through equation

$$y = 0,49 + 0,08\,x \quad \text{for} \qquad 1 \leq x \leq 4$$

If this procedure is applied to similar data (this is not shown) with respect to the scope for other combinations of groups and mass media, we obtain the following equations in Table 6.9.

Table 6.9. Scope of time programming

Newspaper	Group	Equation
X	primary	0,49 + 0,08 x
Y	primary	0,47 + 0,12 x
X	secondary	0,44 + 0,12 x
Y	secondary	0,37 + 0,09 x

For newspaper Y the decrease occurs after 5 inserts. The estimates indicate that both newspapers share the primary group equitably, but newspaper X has 60 % of the secondary group. The management wishes to reach at least 80 % of the primary group (goal 1) and 70 % of the secondary group (goal 2). Furthermore they wish to maintain their previous tradition of having at least twice the numbers of inserts in newspaper X as in Y (goal 3). Newspaper X is paid 3,000 euros per insert. The management has allocated a budget of 16,000 euros for the marketing campaign (budget constraint).

Formulate and solve a goal programming model that allows the company to decide the number of inserts in newspapers X and Y, considering that management places the highest priority on achieving the scope desired for the primary group (goal 1), the second priority on achieving scope desired for the secondary group (goal 2) and the lowest priority on the relationship between the number of inserts in X and in Y (goal 3). In this last goal not reaching the goal is twice as important as exceeding it.

CHAPTER 7

DISCRETE MULTIPLE CRITERIA DECISION MAKING TECHNIQUES

Multiobjective programming and goal programming are applicable to both continuous and discrete problems. In other words, they can be used for making decisions when the number of alternatives is infinite but also when the number is finite and usually small. However, there are a number of methods specifically designed for this latter case that have been applied in many decision problems. In this chapter we will see the Analytical Hierarchy Process (AHP) and the *PROMETHEE* method.

Discrete multiple criteria decision problems can be classified into three groups. First, those in which we want to select only one of the alternatives. For example, which car or mobile phone to purchase, based on economic, technical and design criteria. Another group of problems consists of those in which we want to classify the alternatives, such as grouping the suppliers of a company as good, acceptable and bad, through various attributes (cost, technical, delivery time, etc.). Other problems are those in which our interest is focused on ordering alternatives by priority. For example, to prioritize investment projects for budget allocation.

7.1. ANALYTIC HIERARCHY PROCESS

7.1.1. INTRODUCTION

The Analytic Hierarchy Process, known as AHP, developed by Thomas L. Saaty (1980) has been successfully applied to a wide variety of decision making problems in companies and public administration. Amongst its applications we find strategic planning, resource allocation and selection, market share, production, business ethics, energy, health, education, environment and politics. It is also used to determine the weights in other techniques such as multiobjective and goal programming, as well as PROMETHEE and multi-attribute utility analysis.

As we have seen in the previous chapter, in a **multiple criteria context the optimal concept does not exist**. In general, we can say that the multiple criteria techniques are helpful in the decision making process, which seeks **to integrate the behaviour of the objectives with the judgment** of the decision maker/s, so as to be able to manage and make that subjectivity explicit.

In practice, many decision problems are not presented in a structured way with a list of objectives and alternatives, ready to make a systematic analysis. The choice of the criteria on which we will base our decision is subjective so we must make them explicit and the process transparent. This is particularly important for group or collaborative decision-making. An example would be when you have to select the best suppliers, taking into account the opinion of several people in the company. The AHP method considers the preferences of the decision maker/s through judgments about the relative importance of the criteria and the alternatives taken "in pairs". To apply this approach, quantitative information on the outcome of each alternative on each of the criteria considered is not required, but only the value judgments of the decision maker/s.

7.1.2. BUILDING A HIERARCHICAL MODEL

The first phase of the AHP method consists of building the **decision hierarchy** that represents the multiple criteria decision problem. There are few rules for building hierarchies. The upper level consists of only one element, which is the overall aim of the problem. Successive levels may have multiple elements, all of the same order of magnitude, otherwise they must be at different levels. There is no limit to the number of levels in a hierarchy. If we cannot compare elements in relation to the next level above, one must determine search terms by which they can be compared and put them into an intermediate level. At this stage it is important to identify the actors involved in decision-making, and in particular its objectives and preferences. A widely used technique to design the hierarchy of a decision problem is brainstorming.

Let us take a simple example. A student of Business Administration and Management decides to study their final year degree in another European country through the Erasmus program. After reviewing the possible destinations he/she determines that his/her choices are Aarhus Universitet (Denmark), Universitet Gent (Belgium) and Radboud Universiteit Nijmegen (Holland).

Assuming that the cost of living in different countries is not considered, since the grant already takes that into account, the criteria to consider when making the decision are related to education and leisure. One of the student's objective is to improve his/her English. The prestige of the university is important for his/her Curriculum Vitae, as is the quality of teaching and the number of subjects available in English. Entertainment is also very important because relations with people from different cultures give a lifetime experience. The friendliness of the people, the possibility of going out and travelling through Europe are also relevant issues to the student.

Figure 7.1 shows the decision hierarchy of the student that intends to prioritize the Erasmus destinations that best suit his/her objectives. In general, the approach to building a hierarchy depends on the kind of decision to be made. If it comes to selecting or prioritizing alternatives, we can start from the lower level, choosing the alternatives first. Previous levels would include criteria for evaluating these alternatives and on the upper level there would be a single element, which is the overall goal. Sometimes the hierarchy is designed from top to bottom. In many real problems criteria and objectives that must inform the decision making are not known and the AHP method helps us identify them. There is no limit to the number of levels in a hierarchy. The question to be answered is: "Is it possible to compare the items that are placed on the same level in terms of any of the elements of the next higher level?”. If the answer is no, we should decide in what terms they can be compared and create an intermediate level.

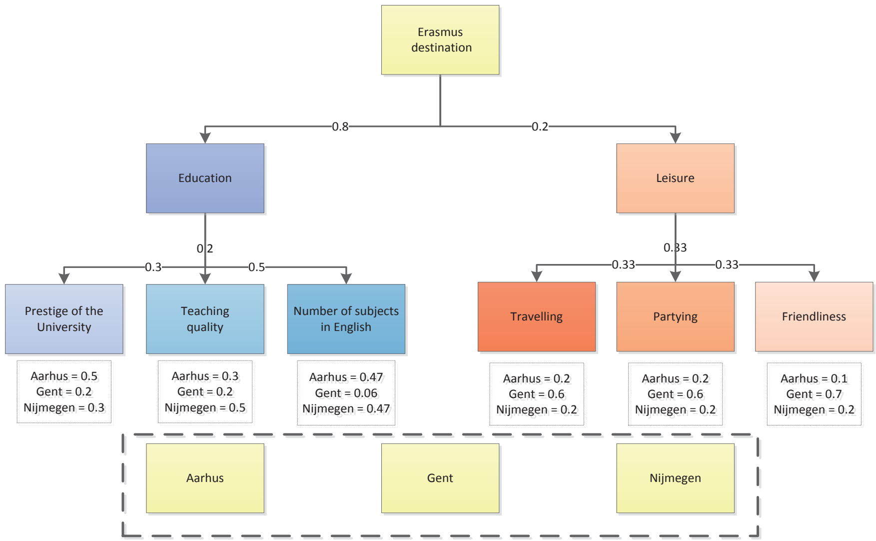


Figure 7.1. Decision hierarchy to prioritize the Erasmus destinations

The student must design the hierarchy that best represents the criteria to be taken into account and alternatives, i.e. all Erasmus destinations that he/she wants to evaluate. One should not consider more than nine destinations, and with not more than nine criteria in the same group in the hierarchy, as it has proven very difficult for people to make pairwise comparisons with such a large number of elements. Another rule to remember is that the elements of the same group must have the same order of magnitude.

7.1.3. SETTING PRIORITIES

In our example, the student's problem is to select in which university he/she should undertake his/her final year of studies. However, the number of places that are offered by the universities is small and other students may also apply for the same destination. Therefore, a prioritized list of the destinations he/she wishes to attend should be made.

The second phase of the AHP method consists of setting priorities between the elements of the hierarchy, then synthesizing our judgments to obtain global priorities to allow us to reach a final decision. For that, we perform pairwise comparisons of elements of the same level regarding the element of the next higher level. In our example, at the second level, the student would ask, how much more important is education compared to leisure when choosing an Erasmus destination? or, on the third level, how much more important is the prestige of the university with respect to the quality of teaching in education?

We can represent the results of pairwise comparisons in a matrix. The matrix is a simple method that enables us to collect information about the judgments and analyze their consistency. To complete the comparison matrix we will use numbers that represent the importance of one element over another. Table 7.1 presents the fundamental scale pairwise comparisons in the AHP method.

Table 7.1. Fundamental scale for pairwise comparisons in the AHP method

Intensity of importance or contribution of one activity over the other	**Definition**	**Explanation**
1	Equal importance	The two elements contribute equally to the objective
2		Intermediate importance between 1 and 3
3	Weak importance of one over another	Experience and judgment slightly favour one element over another
4		Intermediate importance between 3 and 5
5	Essential or strong importance	Experience and judgment strongly favour one element over another
6		Intermediate importance between 5 and 7
7	Demonstrated importance	An element is strongly favoured and its dominance is demonstrated in practice
8		Intermediate importance between 7 and 9
9	Absolute importance	The evidence favouring one element over another is of the highest possible order of affirmation
1/2 1/3 ... 1/8 1/9	If the **first element** has a **strong importance** when compared to the **second element,** we assign a **5 on the scale**. If we make the comparison of **the second element in relation to the first,** the value assigned on the scale is **1/5**	

In our example to compare the three universities located in Aarhus, Gent and Nijmegen, with relation to the number of subjects available in English that we can take, we will fill a 3x3 matrix, whose values are shown in Table 7.2. The criterion "number of courses available in English" is written in the upper left corner and destinations are written in the first row and first column in the same order. This matrix has nine elements and all elements of the main diagonal are 1, since they represent the comparison of each university with itself. Of the remaining six elements, you have to complete only the three judgments above the main diagonal. The judgments below are their reciprocals.

Then students should ask themselves: How much greater is the number of subjects available in English at one university than in another? In Denmark all college degrees are taught in both English and Danish. In Holland college teaching also is performed in English. By contrast, in Gent, Flemish is the main language used in higher education, so the availability of courses in English is lower. Note that the value assigned to an element ij in the matrix is the one resulting from the comparison of the element of row i with the column j. For example, we have assigned a 7 to indicate the comparison between the number of subjects in English from the University of Aarhus and Gent, as there are many more courses available in the first university. If the element of the row is less important than the value of the column its value is a fraction. For example, 1/7 when comparing Gent with Nijmegen.

Table 7.2. Comparison Matrix for Erasmus destinations by number of subjects in English

Number of subjects in English	Aarhus	Gent	Nijmegen
Aarhus	1	7	1
Gent	1/7	1	1/7
Nijmegen	1	7	1

In summary, the matrix of Table 7.2 and any pairwise comparison matrix satisfy the following properties:

1. $a_{ij} \geq 0 \; \forall \; i, j$
2. $a_{ij} = \frac{1}{a_{ji}}, \forall \; i, j$
3. $a_{ii} = 1 \; \forall \; i$

The AHP method requires that each matrix element a_{ij} takes one of the values of the fundamental scale comparisons 1/9, 1/8…1/2, 1, 2, 3… 9.

Once the pairwise comparison matrix has been obtained we can set the relative priorities of the Erasmus destinations on the number of courses available in English. The following method provides a rough estimate of the priorities that we will use in making the decision.

When calculating priorities, we first add the values in each column (Table 7.3). We then divide each cell in each column by the total of the column, in order to obtain a normalized matrix (Table 7.4).

Table 7.3. Calculation of the priorities of the Erasmus destinations by number of subjects in English

Number of subjects in English	Aarhus	Gent	Nijmegen
Aarhus	1	7	1
Gent	1/7	1	1/7
Nijmegen	1	7	1
Column Total	15/7	15	15/7

Table 7.4. Normalized matrix for Erasmus destinations by number of subjects in English

Number of subjects in English	Aarhus	Gent	Nijmegen
Aarhus	7/15	7/15	7/15
Gent	1/15	1/15	1/15
Nijmegen	7/15	7/15	7/15

Finally, we calculate the average of the rows by adding all of the values in each row of the normalized matrix and dividing the sum by the number of cells that are in a row, obtaining **local priorities** of Erasmus destinations for the criterion "number of subjects in English".

$$Aarhus = \frac{{}^{7}/_{15} + {}^{7}/_{15} + {}^{7}/_{15}}{3} = \frac{7}{15} = 0.47$$

$$Gent = \frac{{}^{1}/_{15} + {}^{1}/_{15} + {}^{1}/_{15}}{3} = \frac{1}{15} = 0.06$$

$$Nijmegen \frac{{}^{7}/_{15} + {}^{7}/_{15} + {}^{7}/_{15}}{3} = \frac{7}{15} = 0.47$$

The sum of local priorities should be equal to 1. We follow the same procedure to complete all matrices needed to solve the problem and calculate local priorities of the elements considered in our decision hierarchy. After calculating all local priorities of all pairwise comparison matrices in the hierarchy we can obtain the **global priorities**, the sum must also be equal to 1.

Assuming that local priorities for a student are the values shown in Figure 7.1, we calculate the global priorities of the alternatives weighting their local priorities with the priorities of the objectives. The results are as follows

$$\begin{aligned}
Aarhus = \quad & 0.8 * [(0.3 * 0.5) + (0.2 * 0.3) + (0.5 * 0.47)] + \\
& 0.2 * [(0.33 * 0.2) + (0.33 * 0.2) + (0.33 * 0.1)] = 0.39 \\
Gent \quad = \quad & 0.8 * [(0.3 * 0.2) + (0.2 * 0.2) + (0.5 * 0.06)] + \\
& 0.2 * [(0.33 * 0.6) + (0.33 * 0.6) + (0.33 * 0.7)] = 0.23 \\
Nijmegen = \; & 0.8 * [(0.3 * 0.3) + (0.2 * 0.5) + (0.5 * 0.47)] + \\
& 0.2 * [(0.33 * 0.2) + (0.33 * 0.2) + (0.33 * 0.2)] = 0.38
\end{aligned}$$

Taking into account these global priorities the best Erasmus destination for the student of business administration and management in our example is Aarhus Universitet. The second option would be Nijmegen and lastly Gent.

There are two ways to synthesize the priorities, the **distributive mode** and the **ideal mode**. In the distributive mode the sum of the priorities of the alternatives equals one. It is used when there is dependence between the alternatives and the unit must be distributed between them. However, the ideal mode is the one that should be used when the objective is to select only one of the alternatives. In this case the local priorities of the alternatives are divided by the largest value and this is done for each criterion in such a way that for each criterion there is an alternative that is considered ideal. In both modes the priorities are weighted in the same way with the weights of the objectives. The difference between the two methods is more interesting in theory than in practice.

7.1.4. LOGICAL CONSISTENCY

In the decision making processes it is important to know the consistency of the judgments contained in the comparison matrices, since we do not want our decisions to be based on very inconsistent judgments. For example, it would not be logical that our student knows that the universities of Aarhus and Nijmegen have the same number of courses available in English, and, that when comparing these two universities with Gent, one university has moderate and the other strong importance.

When a matrix is consistent, the average of the sum of each row of the normalized matrix indicates how much the element of the row dominates the others in relative terms. On the other hand, the sum of the columns of the pairwise comparison matrix determines the degree to which each element is dominated by the other elements, so that the product of the two values is equal to one. When a matrix is consistent the elements of any column of the normalized matrix give the same priorities that we obtain by calculating the average of the rows. This means that they are identical to the priorities of pairwise comparison matrix.

Multiplying the sum of the pairwise comparison matrix columns by the average of the sum of each row of the normalized matrix is 1, since they are reciprocal numbers. The sum of all these multiplications is equal to n, if the matrix is consistent. In our example the sum is equal to 3 which represent the three universities considered as alternatives for the destination of the Erasmus student.

As we have seen in the example, the pairwise comparison matrix, **A**, is formed through the comparison of each element to another. If we have n elements (criteria, objectives, alternatives), where their weights or priorities are $w_1, w_2 \ldots w_n$ the pairwise comparison matrix has the following structure:

$$\mathbf{A} = \begin{pmatrix} 1 & w_1/w_2 & \ldots & w_1/w_n \\ w_2/w_1 & 1 & \ldots & w_2/w_n \\ : & : & : & : \\ w_n/w_1 & w_n/w_2 & \ldots & 1 \end{pmatrix}$$

where $a_{ij} = {}^{w_i}/_{w_j}$. This is the relative weight of element i to element j.

The consistency of the judgments is related to the **transitivity of preferences** in the comparison matrix. In summary, a matrix **A** is **consistent** if

$a_{ij} = a_{ik} * a_{kj}, \forall\, i, j, k$.

Other notable properties are as follows:

1. The rank of matrix **A** is 1 as all the rows are multiples of the first row.
2. The eigenvalues of matrix **A** ({λ ∈ R: det (**A** –λ**I**) = 0}) are all zero except for 1 as the rank of the matrix is 1.
3. The trace of matrix A is equal to n, as the diagonal consists of values that are all 1. Therefore, the only nonzero eigenvalue is n, since the sum of the eigenvalues of the matrix coincides with the trace.

4. The eigenvector associated to n coincides with the priority vector

   $\mathbf{w} = (w_1, w_2 \ldots w_n)$. This is $\mathbf{w}$ the nontrivial solution of the system.

   $\mathbf{A}\,\mathbf{w} = \lambda\,\mathbf{w}$

   Only the priority vector whose values sum to 1 is considered.

5. If we use priorities which sum 1, any column of the normalized matrix A coincides with the priority vector.

Priorities are calculated from judgments of pairwise comparison, which are given on a numerical scale. The values of pairwise comparison matrix a_{ij} can be considered as an estimation of the true values of w_i/w_j. In this situation n cannot be an eigenvalue of the matrix and therefore we must find the largest eigenvalue λ_{max} of the obtained matrix.

In summary, a pairwise comparison matrix **A** is consistent if and only if

$\lambda_{max} = n$.

$\lambda_{max} \geq n$ and there is a positive components vector $\mathbf{w}^{\mathbf{max}}$.

There are several methods to obtain the weights vector $\mathbf{w}^{\mathbf{max}}$ and λ_{max}, both exact and approximate. An exact method is calculated directly with the spreadsheet.

The approximate method that we used in the example is based on the property number 5. First we normalize each column of matrix **A**, by adding all values in each column and dividing each cell by the total. Thus we obtain $\mathbf{A}_{norm}$. We then calculate the average of each row of $\mathbf{A}_{norm}$, with the sum of the values in each row and divide this by the number of rows, obtaining in this way the weight vector **w.** We then calculate the product **Aw** and in the last step we calculate λ_{max}.

Inconsistencies can occur for two reasons: One because the decision maker establishes intransitive relationships in pairwise comparisons or because it changes the sense of preference. Although we cannot be so confident in our judgments as to force consistency in the pairwise comparison matrix, we need some degree of consistency in setting priorities for the elements with respect to some criterion to obtain valid results in real applications.

The eigenvalue is used to measure the degree of inconsistency, since, if the comparison matrix is consistent, the largest eigenvalue is equal to n. We define **Consistency Index (CI)** as follows:

$CI = (\lambda_{max} - n)/ (n-1)$

This index represents the cumulative average of inconsistency of the matrix. To find out if it is large or small it is compared to the Random Consistency Index (RCI), which is the average value of CI of pairwise comparisons matrices of the same order randomly obtained. RCI values for the most common sizes of matrices are:

Table 7. 5. Random Consistency Index values for n size of matrices

n	2	3	4	5	6	7	8	9	10
RCI	0.00	0.58	0.90	1.12	1.25	1.35	1.41	1.45	1.51

Inconsistency Index (II) is defined as the ratio of the matrix CI and RCI,

II = CI/RCI.

The consistency of the comparison matrix is considered to be acceptable if the ratio is **less than or equal to 0.10**.

7.1.5. SOFTWARE

We can apply the AHP using a spreadsheet. However there are computer programs with graphics capabilities that allow us to enter, display the results and make sensitivity analysis, e.g. ***Expert Choice*** (expertchoice.com) and ***Super Decisions*** (http://www.superdecisions.com). In particular, the latter software allows us to solve decision problems using AHP and its generalization ANP (Analytic Network Process). ANP is used when the problem cannot be structured hierarchically because there are dependencies and interactions between its components, thus the problem can be best represented by a network than by a hierarchy.

In ***Expert Choice*** once we have defined the decision hierarchy, the software allows us to enter **data** for the comparison matrix **verbally**, **numerically** or **graphically** and it also generates the questionnaire. With *Expert Choice* one can see the necessary degree of detail, the local and global priorities and graphical reports for the sensitivity analysis. In other words, we can look at how the global priorities of the alternatives change when criteria weights change. In the following figures some of the possibilities which are offered by the program are represented using the Erasmus destination example.

In Figures 7.2 and 7.3 we can see the different ways of entering data for our example in *Expert Choice Comparion Suite*. We can introduce data graphically, numerically or verbally. The software generates the questionnaire for pairwise comparisons (Figure 7.2).

Figure 7.2. Entering data graphically/numerically in Expert Choice for Erasmus destination example

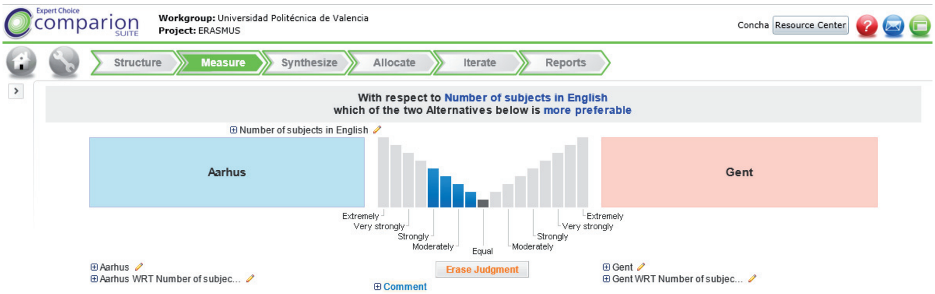


Figure 7.3. Entering data verbally in Expert Choice for Erasmus destination example

In Figure 7.4 we see the results and different types of graphs that can be used to perform sensitivity analysis. For example, by changing the bars that represent the weights that the student gives to the education and leisure criteria, we immediately see the effect on the global priorities of the alternatives. Thus, if we modify the importance that student gives to these two criteria we see how the choice of Erasmus destination would be different, considering the global priorities that are obtained in this case, as seen in Figure 7.5.

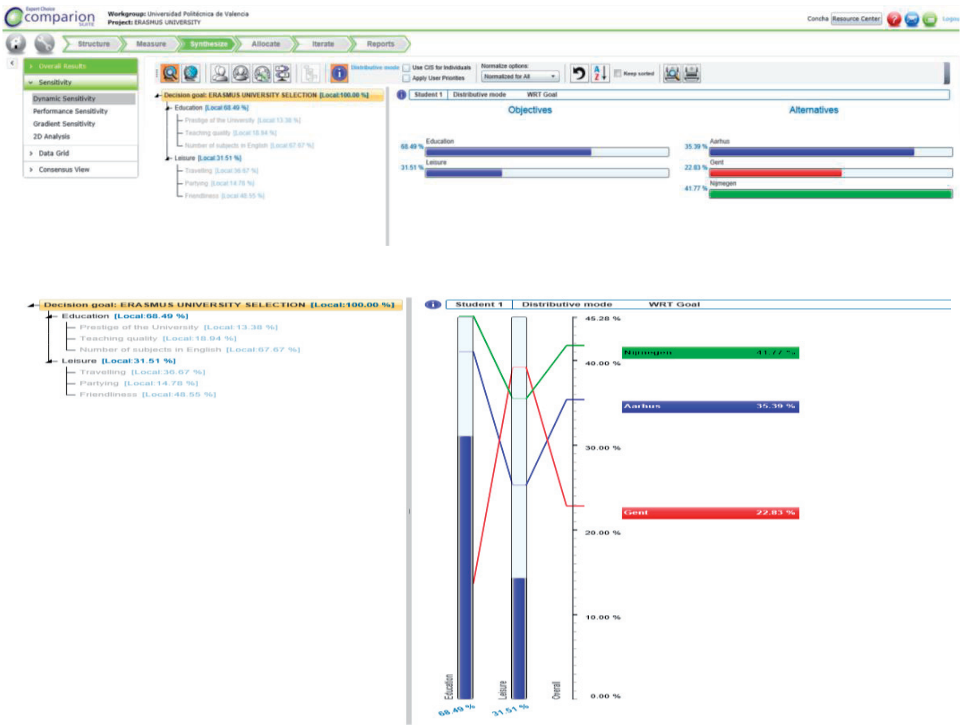


Figure 7.4. Results and sensitivity analysis graphs in *Expert Choice* for Erasmus destination example

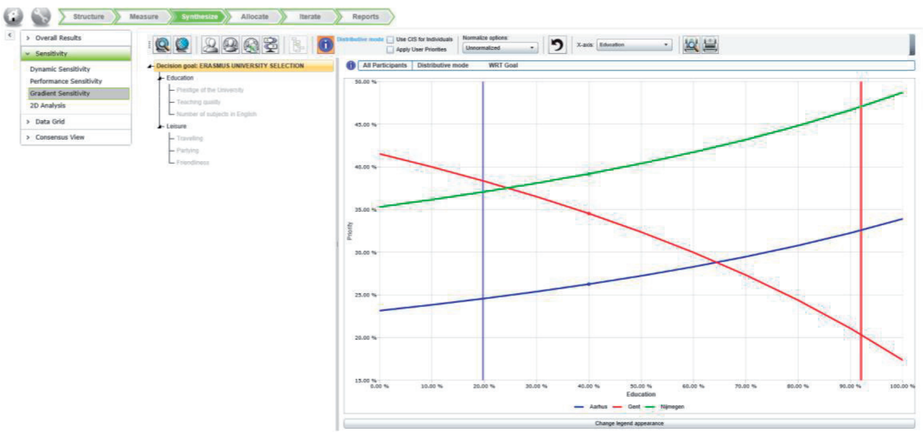


Figure 7.5. Sensitivity Analysis in *Expert Choice* Erasmus for destination example: modification of the weight of education

7.2. PROMETHEE METHOD

7.2.1. INTRODUCTION

Brans developed the PROMETHEE (Preference Ranking Organisation Methods for Enrichment Evaluations) in 1982 and its methodology has been successfully applied in finance, investment planning, industrial location, tourism, hospital management and water management among many other fields.

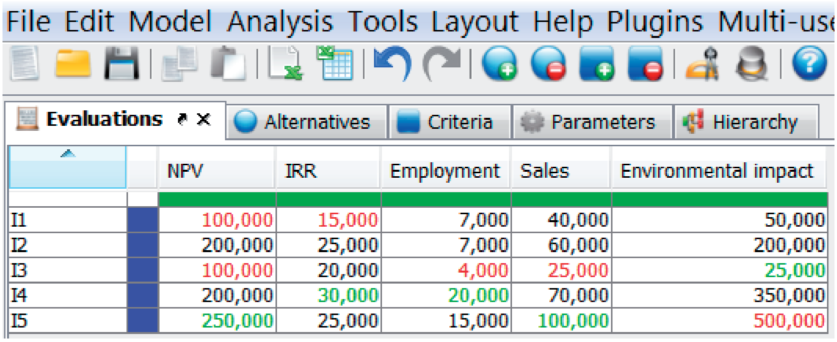
A discrete multicriteria problem is characterized by a finite set of **alternatives** ***(a_1, a_2...a_i… a_n)*** and a set of evaluation **criteria** ***(g_1, g_2...g_i… g_k)***. We can be interested in maximizing some criteria and minimizing others. As we saw in the previous chapter, in general there is no alternative that optimizes all criteria. Suppose you wanted to buy a car. The criteria that may be important for you are the price, design, consumption, security, etc. No car optimizes all criteria at the same time: the cheapest does not imply the least consumption, better security and better design. Thus we need to choose the best compromise solution, which depends not only on the **basic** data, represented in an evaluation table (Table 7.6), but also on our individual preferences. Therefore, we need additional information to represent these preferences.

Table 7.6. Evaluation table

Alternatives	Evaluation criteria					
	g_1	g_2	…	g_j	…	g_k
a_1	$g_1(a_1)$	$g_2(a_1)$	…	$g_j(a_1)$	…	$g_k(a_1)$
a_2	$g_1(a_2)$	$g_2(a_2)$	…	$g_j(a_2)$	…	$g_k(a_2)$
…	…	…	…	…	…	…
a_i	$g_1(a_i)$	$g_2(a_i)$	…	$g_j(a_i)$	…	$g_k(a_i)$
...	…	…	…	…	…	…
a_n	$g_1(a_n)$	$g_2(a_n)$	…	$g_j(a_n)$	…	$g_k(a_n)$

Suppose we have a problem in which we have to prioritize five investments I1, I2 ... I5. The evaluation criteria are the Net Present Value (NPV), Internal Rate of Return (IRR), employment, sales and environmental impact. It is of interest to maximize the first four criteria and minimize the latter.

In Figure 7.6 we present the evaluation table for this problem, which was obtained with D-Sight, the software with which we have made the calculations and presented the results in the remaining figures in this section.



	NPV	IRR	Employment	Sales	Environmental impact
I1	100,000	15,000	7,000	40,000	50,000
I2	200,000	25,000	7,000	60,000	200,000
I3	100,000	20,000	4,000	25,000	25,000
I4	200,000	30,000	20,000	70,000	350,000
I5	250,000	25,000	15,000	100,000	500,000

Figure 7.6. Evaluation table for prioritizing investments

Dominance relations associated with multicriteria problem are defined as follows:

For each pair of **alternatives** *a* and *b*, ***a* is preferred** to ***b*** (***aPb***) if for all considered criteria their value for *a* is equal to or better than their value for *b* and there is at least one criterion for an alternative *a* is better than *b*.

Two alternatives *a* and *b* are **indifferent** (***aIb***) if the value of all the criteria is the same for *a* and for *b*.

Two alternatives *a* and *b* are **incomparable** (***aRb***) if the value of some criterion is better for *a* than *b* and there is at least one criterion that is better for *b* than for *a*.

What dominance relations are there between alternatives in Figure 7.6? The alternatives that are not dominated by any other solutions are **efficient solutions**, a concept we saw in the previous chapter. In real problems many alternatives are incomparable, so we need **additional information** to make the decision. This information can be: *trade-offs* between criteria, weights that indicate the relative importance of criteria, a value function that adds all the criteria in a single function and therefore with a single criterion for which we can find the optimal preferences associated to each pairwise comparison within each criteria, thresholds for the boundaries of preferences, etc.

Many multiple criteria methods have been proposed that need a table like Table 7.6 and they differ in the additional information required. The purpose of the methods is to reduce the incomparability number. All methods should meet a number of requirements such as:

1. They should take into account the size of the difference in the behaviour of the alternatives for each objective: $d_j(a,b) = g_j(a) - g_j(b)$
2. Remove the **scale effect** as the valuation of the criteria is measured in units that may be different.
3. In pairwise comparisons, an appropriate method should inform if an alternative *a* is preferred to *b*, if it is indifferent or if it is incomparable.
4. As each multiple criteria method requires distinct information and calculation procedures, the solutions obtained can be different. Therefore, it is important that the decision makers understand the methods. It is also convenient to analyse the problem using various techniques to propose robust solutions.
5. Methods should provide information about the conflicts between criteria.
6. Most methods require the weights of the relative importance of the criteria. The weights can be assigned directly or using *AHP*. In any case it is desirable to have tools to do sensitivity analyses that allow us to see the impact of the weights in the solution.

Bernard Roy proposed building improvement relationships, enriching the dominance relationships which are based on *ELECTRE* (ELimination Et Choix Traduisant la REalité/ ELimination and Choice Expressing Reality). *PROMETHEE* belongs to a group of multiple criteria methods, known as *Outranking* methods. In both cases there are several versions of the method, depending on the type of problem to be solved (*PROMETHEE I, II, III, IV, V, VI*).

7.2.2. INFORMATION FOR PREFERENCES MODELLING

In addition to an evaluation table, *PROMETHEE* requires information on the weights of the relative importance of the criteria, we call these $w_1, w_2 \ldots w_k$. The sum of all is the unit:

$$\sum_{j=1}^{k} w_j = 1$$

The preference structure of *PROMETHEE* is based on pairwise comparisons, as in *AHP*. However, in this case comparisons are based on the difference between the valuations of two alternatives of a particular criterion. The larger the difference between evaluations of the alternatives, the greater is the preference for the alternative which behaves better. When the difference is small, the decision maker can consider it to be negligible. These preferences are represented by real numbers between 0 and 1. Figure 7.6 shows that investment I5 is 50 units better than I4 and 150 units better than I1 for NPV.

In general, for each criterion j, the decision maker has the preference function between *a* and *b*, $P_j(a,b)$, a function that depends on the difference between the behaviour of the alternatives:

$P_j(a,b) = F_j[d_j(a,b)]$ for all alternatives of the problem, where

$d_j(a,b) = g_j(a) - g_j(b)$ and

$0 \leq P_j(a,b) \leq 1$

In case the criterion is maximized, the preference will be *a* to *b* for criterion evaluations and the preference can have the shape of Figure 7.7. When the deviation is negative, the preference is 0, and therefore

if $P_j(a,b) > 0$ then $P_j(b,a) = 0$

When we want to minimize the criterion it would be the other way round or we consider the preference function as follows:

$P_j(a,b) = F_j[-d_j(a, b)]$

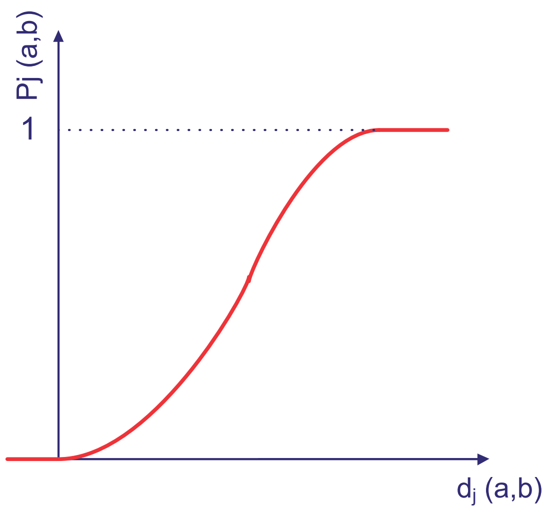


Figure 7.7. Preference function

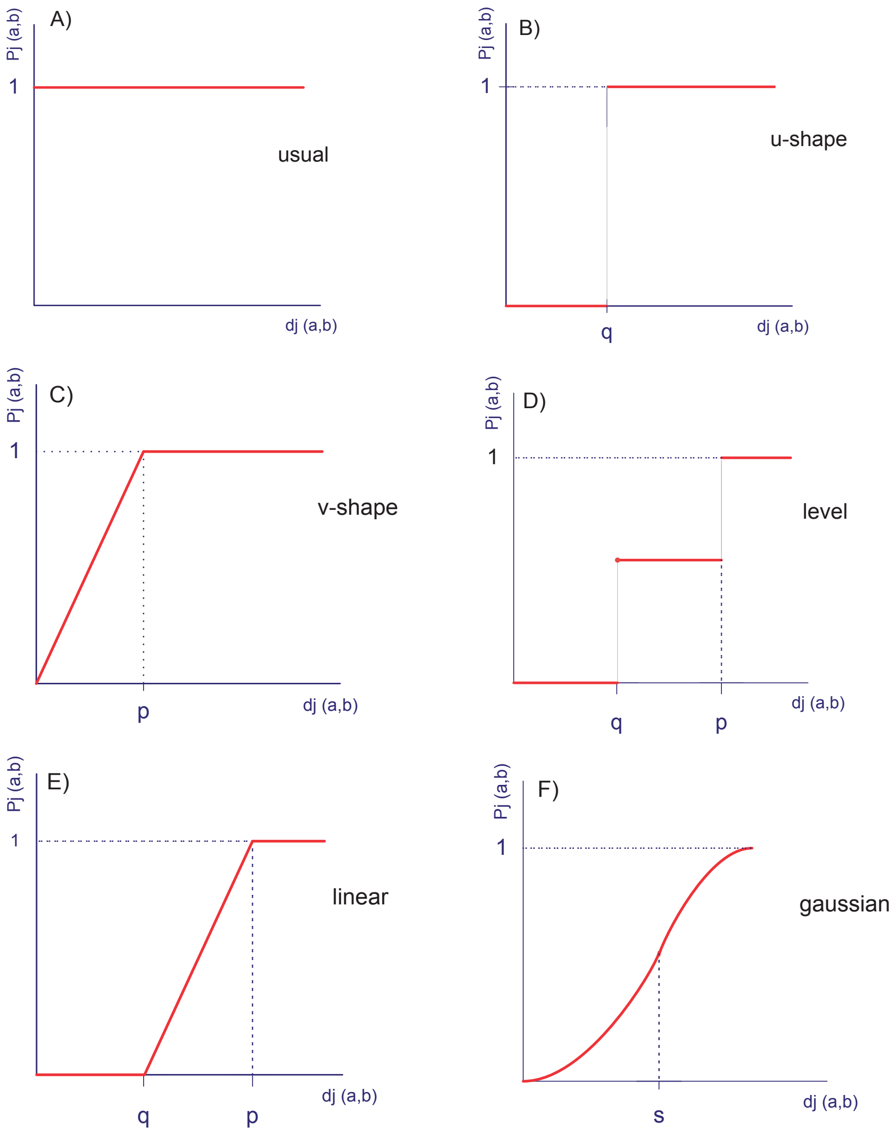


Figure 7.8. D-Sight Preference Functions

For each criterion, we must propose a preference function. The most common functions are those presented in Figure 7.8, and they are the ones that you can choose from the D-Sight software. But in the former case, *usual* (A), where the preference is 1 if the difference between *a* and *b* is positive, and zero otherwise. In other cases we have to define one or two parameters. We call **q** the **indifference threshold**, **p** the **strict preference threshold** and **s** is an intermediate value between them. q is the value of the largest deviation that the decision maker considers negligible, while the preference threshold p is the smallest value of the deviation between alternatives to be considered sufficient for a strict preference of one alternative to another. In case B parameter q must be set, which is p in case C and, in cases D and E the two parameters must be set, the indifference threshold and strict preference threshold. And in the Gaussian preference function (F) s must be set, which will be between p and q. In all cases, when the difference in the behaviour of the alternatives is negative, the preference will be zero (Figure 7.8).

Figure 7.9 shows that we have chosen a type of linear preference function for all objectives, indifference thresholds and preference for each case and the weights of the criteria.

Evaluations | Alternatives | Criteria | **Parameters** | Hierarchy

Criteria		Type	Min/Max	Function	Abs/Rel	Indiff.	Pref.	Weight	Unit	Scale	Decimals
NPV		Pair Wise	Maximize	Linear	Absolute	0	50	25,0%		Numerical	3
IRR		Pair Wise	Maximize	Linear	Absolute	0	5	25,0%		Numerical	3
Employment		Pair Wise	Maximize	Linear	Absolute	0	3	20,0%		Numerical	3
Sales		Pair Wise	Maximize	Linear	Absolute	0	15	10,0%		Numerical	3
Environmental impact		Pair Wise	Minimize	Linear	Absolute	0	100	20,0%		Numerical	3

Figure 7.9. Preference functions and parameters

7.2.3. PROMETHEE I AND II

To apply the method we need to know the evaluation table, the weights of the criteria and the preference functions. First we define the **Aggregated Preference Indices** for each pair of alternatives *a* and *b*:

$$\pi\ (a,b) = \sum_{j=1}^{k} Pj(a,b)w_j$$

$$\pi\ (b,a) = \sum_{j=1}^{k} Pj(b,a)w_j$$

Where ***π (a, b)*** expresses **the degree to which the alternative *a* is preferred over *b*** and *π (b, a)* the degree to which the alternative *b* is preferred over *a*. In most cases there are criteria for which *a* is preferred to *b* and others for which b is preferred to the alternative a. These indices have the following properties:

π (a, a) = 0

0 *≤ π (a, b)* ***≤ 1***

0 *≤ π (b, a)* ***≤ 1***

0 *≤ π (a, b) + π (b, a)* ***≤ 1***

If the aggregated preference index of *a* over *b*, *π(a, b)* is close to zero, then there is a weak global preference for *a* over *b,* and if it is close to 1, there is a strong global preference. Figure 7.10 shows the aggregated preference indices for the investments example.

Thus, using the data in Figures 7.6 and 7.9 (weights of objectives and preference functions) we obtain

$\pi\ (I5, I2) = \sum_{j=1}^{k} Pj(I5, I2)w_j = 1*0.25+0*0.25+1*0.2+1*0.1+0*0.2 = 0.55$

$\pi\ (I4, I2) = 0*0.25 + 1*0.25 + 1*0.20 + 0.666*0.10 + 0*0.20 = 0.5166$

Calculate the aggregated indices for other pairs of investments. Check the results in Figure 7.10. Analyse the preferences of some investments over others as well.

As each alternative is compared with the other (n-1) positive and negative outranking flows are defined. ***Positive outranking flow*** expresses to what extent an alternative outranks all the others. The higher the positive outranking flow, the better the alternative. The value of this expresses the strength of the alternative.

$$\varphi^+\ (a) = \frac{1}{n-1}\sum_{x\in A}\pi\ (a, x)$$

In contrast, the ***negative outranking flow*** indicates to what extent an alternative is overcome by the other. It is therefore an indicator of weakness and the alternative is better when its negative flow is smaller.

$$\varphi^-\ (a) = \frac{1}{n-1}\sum_{x\in A}\pi\ (x, a)$$

For the investment example

$$\varphi^+\ (I1) = \frac{1}{4}(0.2 + 0.3 + 0.2 + 0.2) = 0.225$$

$$\varphi^-\ (I1) = \frac{1}{4}(0.6 + 0.3 + 0.8 + 0.8) = 0.625$$

Calculate positive and negative flows of other investments. Check the results in Figure 7.11.

Pref. Degree Table

	I1	I2	I3	I4	I5
I1	0,000	0,200	0,300	0,200	0,200
I2	0,600	0,000	0,800	0,200	0,200
I3	0,300	0,200	0,000	0,200	0,200
I4	0,800	0,517	0,800	0,000	0,650
I5	0,800	0,550	0,800	0,350	0,000

Figure 7.10. Aggregated preference indices between alternatives

In the ***PROMETHEE I*** we obtain the partial ranking P^I, I^I *and* R^I of the alternatives from the **positive and negative flows**. We do not usually get the same rankings from the two kinds of flows.

Alternative *a* is preferred to *b*, ***a*** $\boldsymbol{P^I}$ ***b***, if *a* positive flows are greater than *b,* and the lower negative or the positive of *a* and b are equal, or *a* lower negative or a positive flows are higher than *b* and the negatives are equal in both alternatives.

Alternative *a* is indifferent to *b*, ***a*** $\boldsymbol{I^I}$ ***b*** if both positive and negative flows are equal in *a* and *b*.

Alternative a is incomparable to *b*, ***a*** $\boldsymbol{R^I}$ ***b***, if *a* positive flows are greater than *b* and the negative also or *a* positive flows are lower than *b* and the negative are also lower than *b*. This usually happens when the alternative *a* is good for a group of criteria where *b* is weak and the alternative *b* is good for criteria in which *a* is not.

Net Flow Table

Alternative	Rank	Score	Positive Flow (Pairwise)	Negative Flow (Pairwise)
I1	4	-0,400	0,225	0,625
I2	3	0,083	0,450	0,367
I3	5	-0,450	0,225	0,675
I4	1	0,454	0,692	0,238
I5	2	0,312	0,625	0,312

Figure 7.11. Positive, negative and net flows of the alternatives

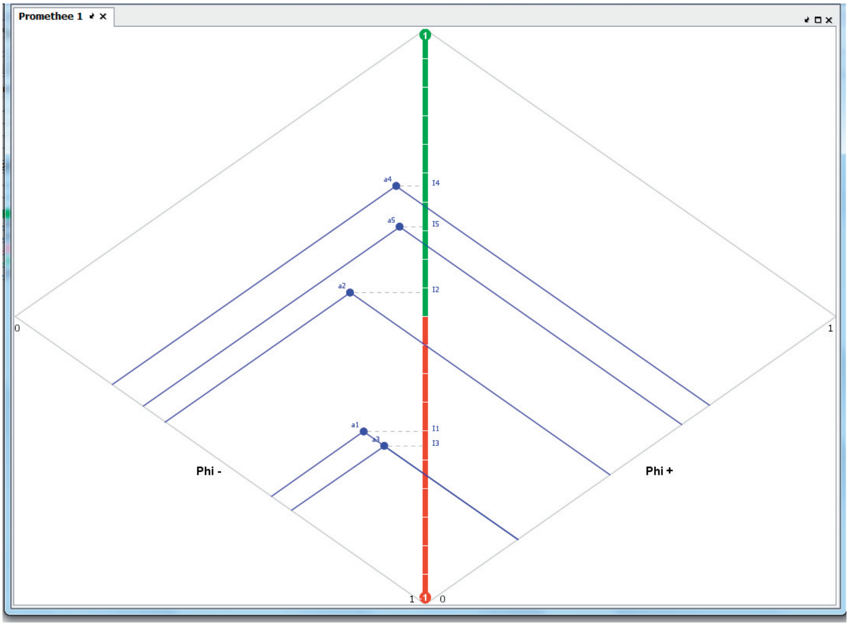


Figure 7.12. Representation of the partial ranking (positive and negative flows)

Figure 7.13. Profile of alternatives Investment 4 (I4) and Investment 5 (I5)

In Figure 7.14 we have the matrix of net flows for all alternatives of the decision problem that is obtained when the whole weight (100%) is given to a criterion.

Unicriterion Matrix

	NPV	IRR	Employment	Sales	Environmental impact
I1	-0,750	-1,000	-0,250	-0,500	0,688
I2	0,250	0,250	-0,250	0,083	0,000
I3	-0,750	-0,500	-1,000	-1,000	0,812
I4	0,250	1,000	1,000	0,417	-0,500
I5	1,000	0,250	0,500	1,000	-1,000

Figure 7.14. Net flows matrix

The PROMETHEE II provides a complete ranking of alternatives (**P^{II}, I^{II}**). In this case we define net flow of alternative *a* as the balance between positive and negative flows:

$$\varphi(a) = \varphi^{+}(a) - \varphi^{-}(a)$$

The higher the **net flow,** the better the alternative and therefore, in this method, **all alternatives are comparable**. The values of net flows of the alternatives are between -1 and 1 and the sum of all of them is 0. If the net flow of **alternative *a*** is **positive**, it is better than all alternatives for all criteria and when the net flow is negative then it is worse than the other alternatives. See in Figure 7.11 the values of net flows in score column for investment example.

PROMETHEE II is easy to use, but the incomparability analysis can help us make decisions in real problems. As the net flow gives us full ranking, it may be compared with a utility function. Figure 7.15 shows the net flow of each investment, which provides us with the ranking of the alternatives.

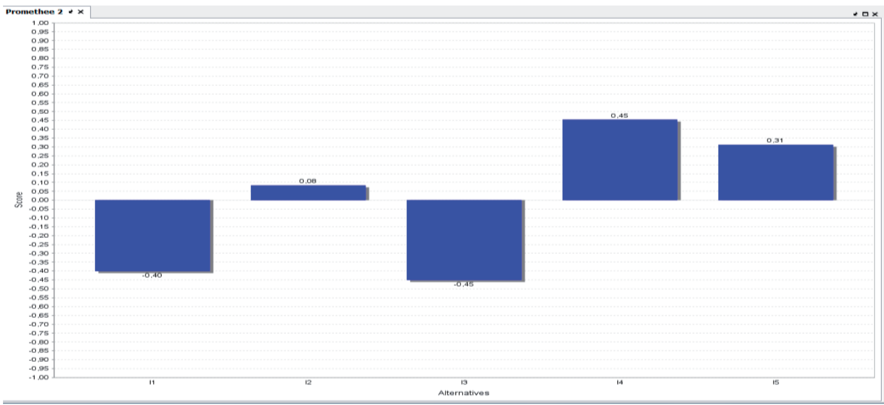


Figure 7.15. Net flows of investment alternatives

The net flow of the alternative *a* for criterion j, **φ_j (a),** is obtained when only this criterion is considered, i.e. 100% of the weight is given to that criterion. This concept expresses how this alternative outranks the other alternatives for this criterion (**φ_j (a) > 0**) or how it is outranked by the other alternatives for criterion j (**φ_j (a) < 0**). The profiles of the alternatives show their quality in all considered criteria. Figure 7.13 presents profiles for I4 and I5 investments. We can see how alternative I5 has the greater NPV and sales than I4; however the negative environmental impact is greater in I5 than in I4.

The net flow of an alternative is the scalar product between the vector of weights and the profile vector of this alternative. This property is used to construct the GAIA plane, which is a graphical tool for analysing multiple criteria problems (Global Visual Analysis, **GVA**) and can be seen in Figure 7.16. The alternatives are represented as points and the criteria are the axes. The red axis is the decision. The GAIA plane shows the discriminating power of the criteria, the conflicting aspects and the quality of the alternatives on different criteria. This plane has a number of interesting properties to interpret the results. For example, the longer the axis of a criterion the more discriminant is that criterion. The criteria with similar preferences have axes oriented approximately in the same direction. The conflict criteria are oriented in opposite directions. The criteria that are not related to others in terms of preferences are represented by orthogonal axes. Similar alternatives are represented by close points and good alternatives with regard to a criterion are represented by points located in the direction of the axis of this criterion (see Figure 7.16).

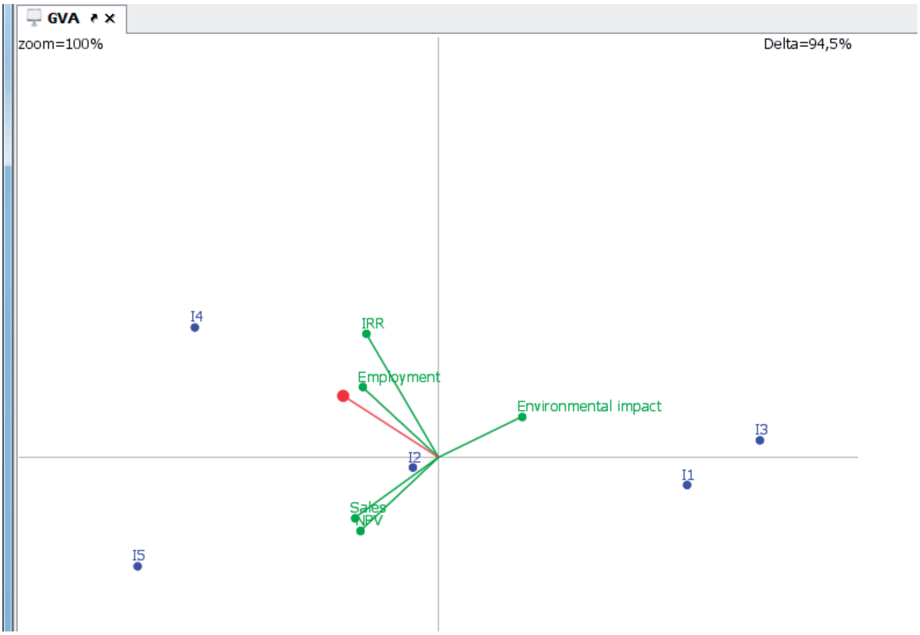


Figure 7.16. GAIA drawing

The D-Sight software provides a variety of graphics that we can use as a tool for analysis and presentation of results as Figures 7.17 and 7.18. The *spider web* graph represents multiple profiles of alternatives, criterion by criterion. The centre of the graph represents the value -1 and the end point of the axes represents +1 (Figure 7.17). It also allows sensitivity analysis such as that in Figure 7.19, which shows that the best choice is insensitive to the weights of the criteria.

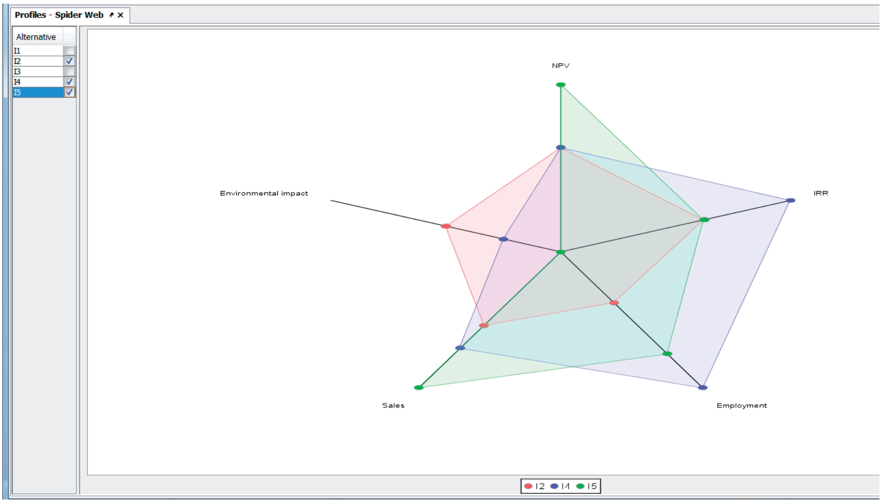


Figure 7.17. Spider web graph of the selected alternatives (I2, I4 e I5)

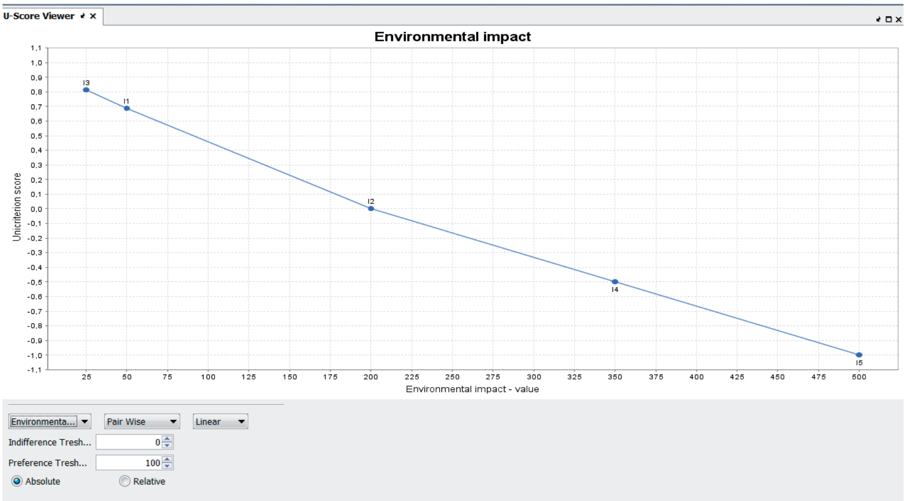
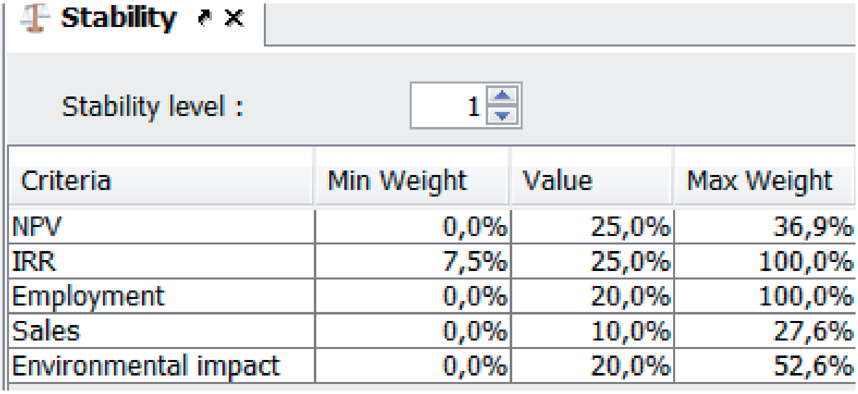


Figure 7.18. Graphical representation of the alternatives for the environmental impact


Stability

Stability level : 1

Criteria	Min Weight	Value	Max Weight
NPV	0,0%	25,0%	36,9%
IRR	7,5%	25,0%	100,0%
Employment	0,0%	20,0%	100,0%
Sales	0,0%	10,0%	27,6%
Environmental impact	0,0%	20,0%	52,6%

Figure 7.19. Sensitivity analysis

7.3. COLLABORATIVE DECISION MAKING

We can also apply AHP when the decision is made in a group. In this case we can distinguish two fundamental aspects. Firstly, how to add individual preferences into a collective judgment and secondly, how to build a group preference from individual preferences.

If a group of people are involved in making decisions, we have to determine consensus matrices and local and global weights or priorities that represent the preferences and priorities of the group. We can use different techniques to obtain them. One of the most used technique is **the geometric mean of all pairwise comparisons**, as defended Saaty and others (Saaty and Peniwati, 2008; Xu, 2000). Saaty considers that the geometric mean is necessary because of comparison between two elements j and i, $a_{ji,}$ should give us the reciprocal value assigned to a_{ij} the original comparison. The arithmetic mean does not satisfy this reciprocal relationship. Moreover, given a group of inconsistent individuals, the inconsistency of the group by adding individual judgments by the geometric mean is at most equal to the largest individual inconsistency. Another alternative to add preferences or judgments and to obtain consensus matrices in group decision making is to use the goal programming models developed by González-Pachón and Romero (2004, 2007).

Both AHP and goal programming models are suitable techniques for group decision making. However AHP has several advantages. The first is the simplicity of the method, easily understandable by all members of the group who do not need to be experts in decision making techniques. The second advantage is the use of the geometric mean for adding preferences/judgments of pairwise comparison, where we always obtain consistent consensus matrices from consistent individual matrices. However, one of the most frequent criticisms of AHP is that when adding a new option, the ranking can change (rank reversal). Saaty usually argues by saying that it would be a different decision problem.

The goal programming models developed by González-Pachón and Romero have an advantage compared with AHP, they do not require that preferences or individual judgments are consistent. However, these models have the drawback that the consensus matrix obtained by adding preferences can be inconsistent in theory. It is possible to develop a model to avoid this, but a nonlinear goal programming model would have to be solve. Also note that the method is difficult for participants without previous training in optimization techniques to understand.

In Annex 3 we explained, with an example, how to use the *Expert Choice Comparion Core* software for collaborative decision making that allows us to aggregate the preferences of a working group and schedule sessions for making collective decisions in real time. Finally, note that we can also use *PROMETHEE* for collaborative decision making. For this purpose there is D-Sight software web platform.

7.4. SUMMARY

In this chapter we have seen two widely used methods to solve multiple criteria decision problems when we have a small number of decision alternatives. *AHP* is based on designing a decision hierarchy that represents the problem and allows the weights of objectives and priorities of alternatives from the pairwise comparison of elements to be obtained. In this case, the decision-maker expresses value judgments about the importance or preference of one element over another, using the Saaty scale. The judgments must be consistent in order to select or prioritize the alternatives. *Expert Choice* software and *Super Decision* help address the multiple criteria decision problems, both individually and at the group level. *PROMETHEE* is another method that is also based on pairwise comparison of alternatives. However, in this case it is necessary to know the behaviour of the alternatives for each criterion quantitatively or qualitatively. In this method the preferences of the decision maker are incorporated through preference functions. This approach compares the alternatives by using the net flow, which is an independent concept of the units in which the criteria are measured. D-Sight is a tool for solving discrete multiple criteria problems with the PROMETHEE method for individual and collaborative decision making using a web platform.

7.5. SELECTED REFERENCES

1. Belton, V. and Stewart, T.J. 2003. Multiple Criteria Decision Analysis— an integrated approach. Kluwer Academic Publishers, Dordrecht, The Netherlands
2. Brans, J.P. and Mareschal, B. 2005. PROMETHEE Methods, chapter 5 en Multiple Criteria Decision Analysis. State of the Art Surveys. Edited by Figuera, J.; Greco, S. y Ehrgott, M. Springer. Pp.163-195.
3. Expert Choice Comparion Core.2013. www.expertchoice.com.
4. D-Sight. 2013. http://www.d-sight.com.
5. González-Pachón, J. and Romero, C. 2004. A method for dealing with inconsistencies in pairwise comparisons, European Journal of Operational Research, 158 (2), 351–361
6. González-Pachón, J. and Romero, C. 2007. Inferring consensus weights from pairwise comparison matrices without suitable properties, Annals of Operational Research, 154 (1), 123–132.
7. Saaty, T.L. and L.G. Vargas 2001. Models, methods, concepts & applications of the Analytic Hierarchy Process. Kluwer Academic Publishers.333pp.
8. Saaty, T.L. 2006. Fundamentals of decision making and priority theory with the analytic hierarchy process. RWS Publications, Pittsburgh, USA. 478 pp.
9. Saaty, T.L. 2008. Decision making with the analytic hierarchy process, Int. J. Services Sciences, 1 (1), 83-98.
10. Saaty, T.L. and Peniwati, K. 2008. Group Decision Making: Drawing out and Reconciling Differences. RWS Publications. 385 pp.
11. Super Decisions. 2012. http://www.superdecisions.com.

7.6. CASE STUDIES

CASE STUDY 1: SELECTION OF ERASMUS/LEONARDO DESTINATION

Each student must design a hierarchy that represents the personal criteria and objectives to consider in the destination selection where you would attend a year or a semester at another university, either within the Erasmus, Leonardo or any other programme of your interest. The case study can be done with *Excel* and/or *Expert Choice*.

1. Carry out the hierarchy that represents the criteria and objectives you would consider at your personal level.
2. Obtain the matrices of pairwise comparisons.
3. Calculate local and global priorities.
4. Carry out the sensitivity analysis.
5. Write a brief report to support the choice.
6. Choose a destination for a group of friends through collaborative decision making.

CASE STUDY 2: PREFERENCE AGGREGATION AND CALCULATION OF PRIORITIES THROUGH GOAL PROGRAMMING

González-Pachón and Romero (2007) have developed a goal programming model for obtaining the consensus matrix, for example for a group of students that choose the same Erasmus destination.

Decision variables of the model that allows us to obtain the consensus matrix for a group are:

R_{ij}^{C}= the consensus value ratio that quantifies the aggregated judgment when comparing the criterion/alternative i with j.

N_{ij}^{K} *and* P_{ij}^{K}= positive and negative deviation variables from the goal when the student K compares the criterion/alternative i with j

Upper and lower limits of consensus ratios due to the application of the fundamental Saaty scale:

$$0.111 \leq R_{ij}^{C} \leq 9 \qquad\qquad i,j = 1,2,\dots,n$$

In the case that the consensus ratio between two criteria and the value of ratio of the student K are different, this difference will be positive or negative deviation variables as indicated by the model goals.

$$R_{ij}^{C} + N_{ij}^{K} = R_{ij}^{K} + P_{ij}^{K} \qquad i,j = 1,2,\dots,n \quad i \neq j \qquad K = 1,2,\dots,m$$

The achievement function is to minimize the sum of the deviation variables for all students in the group.

$$MIN \sum\sum\sum \left(N_{ij}^{K} + P_{ij}^{K}\right) \qquad i,j = 1,2,\dots,n \quad i \neq j \qquad K = 1,2,\dots,m$$

In a second phase we obtained the weights of relative importance that the group of students i give to the criterion r from the consensus matrix obtained with the previous model, using another goal programming model also developed by González-Pachón and Romero (2004).

In the latter model $\boldsymbol{W_{ri}}$ **decision variables** are the weights of the student group i for criterion r. In the same way as for the previous model, we define the variables of positive and negative deviation for goals and student group i (i = 1, 2... m).

$$W_{r}^{i} + N_{r,s}^{i} = R_{r,s}^{Ci} \cdot W_{si} + P_{r,s}^{i} \qquad r,s = 1,2,\dots,n \quad r \neq s \qquad i = 1,2,\dots,m$$

It is also necessary to add the constraint that the sum of all weights is 1 for each group of students i:

$$\sum_{r=1}^{r=n} W_{r}^{i} = 1 \qquad r = 1,2,\dots,n$$

The achievement function is to minimize the sum of all positive and negative deviation variables for all criteria.

$$MIN \sum\sum\sum \left(N_{r,s}^{i} + P_{r,s}^{i}\right) \qquad r,s = 1,2,\dots,n \quad r \neq s$$

1. Develop the goal programming models that address the problem of group decision making of the case study 1 and obtain the solution.
2. Compare the results obtained from applying the AHP method.

CASE STUDY 3: CAR SELECTION

Develop a multiple criteria model that allows you to choose which car to buy from the five models on the market that are the most interesting to you. Please consider at least the criteria of price, consumption, pollution and design.

1. Develop a model, obtain the necessary data and the solution by the *AHP* method.

2. Solve the problem by using the *PROMETHEE* method. Write a report comparing the data required and the solution obtained with those from the *AHP* method.

CASE STUDY 4: PHONE SELECTION

Develop a multiple criteria model that allows you to prioritise which phone to buy taking into account several criteria. Obtain the necessary data and the solution by using *AHP* and PROMETHEE methods. Finally, write a report comparing the data required and the solution obtained with both methods.

CHAPTER 8

NONLINEAR PROGRAMMING

In previous chapters we have dealt with decision making in a great variety of problems by building and solving linear programming models, using Excel spreadsheets or LINGO optimization software. In general, optimization models are no more than an approximation to the real problem as we explained in chapter 1. Thus, in production planning with the objective of minimizing variable cost, costs per unit were considered constant, independent of production levels. In other words, we did not take into account the situation of increasing marginal costs. Nevertheless, in integer programming we saw how to express a certain type of nonlinearity in a linear form by using binary variables. For example the discounts applied to the acquisition of raw materials depending on the amount bought.

Nonlinear programming occurs when the objective function and/or some or all the constraints in a mathematical model are nonlinear. Nonlinear models are used to represent real problems from a wide range of fields, such as financial engineering or mixes in chemical processes. There are also problems in which we have to find the values of the variables that optimize a given nonlinear function. Applications in business administration and management appear in resource allocation, investment portfolio selection and inventory models, amongst others.

This chapter first presents the main characteristics of nonlinear optimization methods which are relevant for building and solving models for business administration and management. Secondly, we deal with some applications by building and solving models using *Solver* from Excel, with special emphasis on models of investment portfolio selection by Markowitz and Sharpe, Nobel Prize for economy in 1990, which are the basis of modern financial theory taking into account asset returns and risk.

8.1. INTRODUCTION: BASIC CONCEPTS

Nonlinear programming solving techniques use some concepts from differential calculus and algebra. Let us refresh some basic concepts which are necessary to understand this chapter.

Function f in figure 8.1 has local maxima in X_0, X_3 and a, where X_3 is the global maximum. The local minima are X_1, X_2 and b, where X_2 is the global minimum.

Observe in figure 8.1 that on the right hand of X_0 the curve slope at a point is positive and increasingly smaller. By contrast, on the right hand of X_0 the slope is negative. At point X_0 the slope is horizontal or zero. In mathematical terms, the curve slope of a function f is given by its first derivative [d f(x)/ dx = f'(x)]. Therefore, in X_0 the derivative must be zero. Using a similar procedure, we can see that if the function has a local or global minimum at point X_2 its derivative is also equal to zero.

a) Local and global maximum and minimum

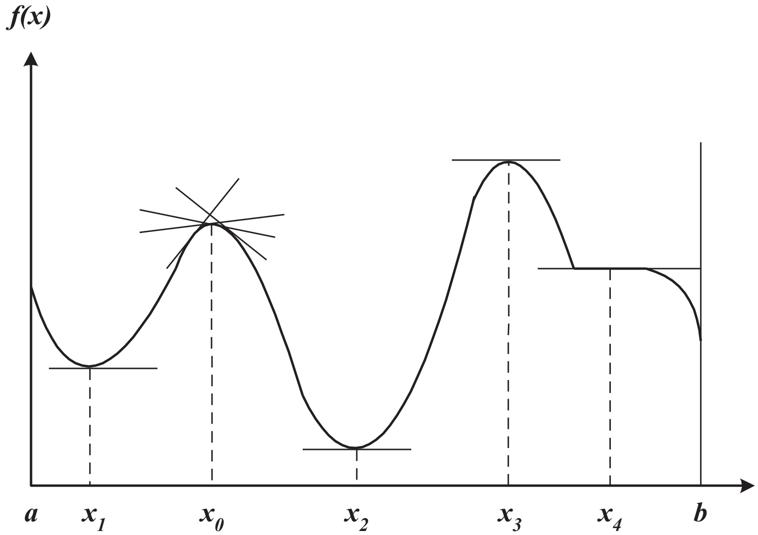


Figure 8.1. Local and Global maximum and minimum

A necessary condition for function f to have a maximum or minimum value in an interior point X_0 of interval [a, b] is that X_0 be a stationary point, i.e., $f'(x) = 0$ at X_0.

Unfortunately, the function has a stationary point at X_4, which is an inflection point. Therefore, the fact that the first derivative is eliminated at one point is not a sufficient condition for that point to be a maximum or a minimum. The sufficient condition for a stationary point to be a maximum or minimum value can be obtained by examining Taylor's development of f around X_0 for small h:

$f(X_0 + h) = f(X_0) + h\, f'(X_0) + h_2/2\, f''(X_0) + R_2$

where $f'(X_0)$ is the first derivative evaluated in X_0, $f''(X_0)$ the second derivative evaluated in X_0 and, for small h, the remaining term R_2 is lower, in absolute value, than the term containing $f''(X_0)$.

At the stationary point X_0, the term containing $f'(X)$ is zero and h_2 is positive for all values of h, negative or positive. Therefore, f (X0 + h) will be lower than $f(X_0)$ if $f''(X_0) < 0$ (X_0 will be a local maximum) and higher than $f(X_0)$ if $f''(X_0) > 0$ (X_0 will be a local minimum).

If X_0 is a stationary point, function f has a maximum at X_0 if $f''(X_0) < 0$ and a minimum if $f''(X_0) > 0$.

If $f''(X_0) = 0$, the first derivative of higher level different from zero at X_0 is found. If the order of the derivative is odd, f has an inflection point at X_0. If the order of the derivative is even it is replaced by $f''(X_0)$ in the previous sufficient condition.

Observe that neither the sufficient nor the necessary condition allow determining whether a stationary point will be a local or global extreme value of function f. It will have to be determined by evaluating the function in all the stationary points of interest, as well as points *a* and *b* of the interval for X.

b) Convex and concave functions for one variable

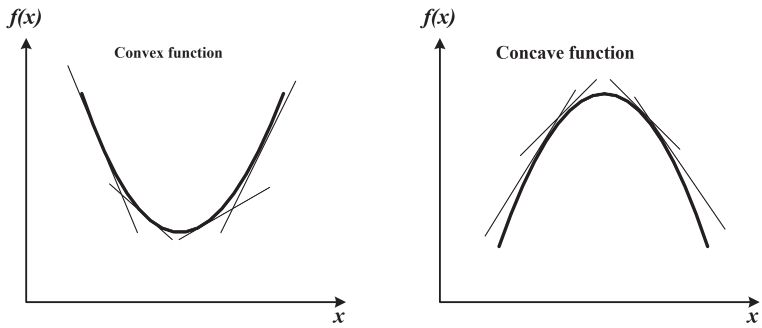


Figure 8.2. Convex and concave functions

Function f of variable x is convex if the line between two arbitrary points on the graph lies above or on the graph of the function. The function is strictly convex if the line between any two points always lies above the function graph. A function f is concave if -f is convex. A linear function is both concave and convex. The second derivative of a function represents the change of the function slope. It is easy to see graphically that:

if $f''(x) \geq 0$ for every x, f is convex

and if $f''(x) \leq 0$ for every x, f is concave

Observe the analogy between these two properties and the sufficient conditions for *a* extreme point of f.

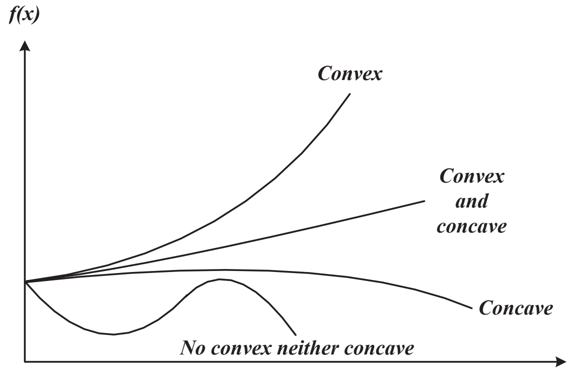


Figure 8.3. Convex and concave functions

8.2. CHARACTERISTICS OF NONLINEAR OPTIMIZATION METHODS

The most important characteristic of nonlinear programming methods is that there is no method that can be considered as "the best" to solve all types of nonlinear problems. Many different algorithms have been developed to solve particular types of nonlinear problems, but none of them can be compared to the simplex method for nonlinear programming, in the sense that it is a very efficient algorithm that allows one solve any type of problem formulated in linear programming form.

Why are nonlinear problems more difficult to solve than linear problems? The optimal solution can not lie at a corner point of the feasible region. For example, suppose that the profit of a company is

$$B = 750 - 0.1\,(X_1 - 50)^2 - 0.2\,(X_2 - 50)^2$$

where X_1 and X_2 are the quantities of the two products it manufactures. The firm wants to maximize this profit taking into account its resource constraints. Thus, the model will be:

$$\text{Max} \quad 750 - 0.1\,(X_1 - 50)^2 - 0.2\,(X_2 - 50)^2$$

with the constraints

$$X_1, X_2 \geq 0$$

$$5\,X_1 + 1\,X_2 \leq 200 \text{ (work force)}$$

$$1\,X_1 + 2\,X_2 \leq 90 \text{ (machinery)}$$

Figure 8.4 shows that the optimal solution lies in a point of the machinery constraint that is not a corner point.It can also be the case that the optimal solution lies inside the feasible region. This would occur in this problem if the work force available was 400 and the machinery were 200.

Another drawback of nonlinear optimization methods is that they generally find a local optimum but not the global optimum. All linear programming methods (simplex), as well as those of integer programming generate a global optimum. Finally, another difficulty arises from the fact that nonlinear constraints can generate non-convex feasible regions.

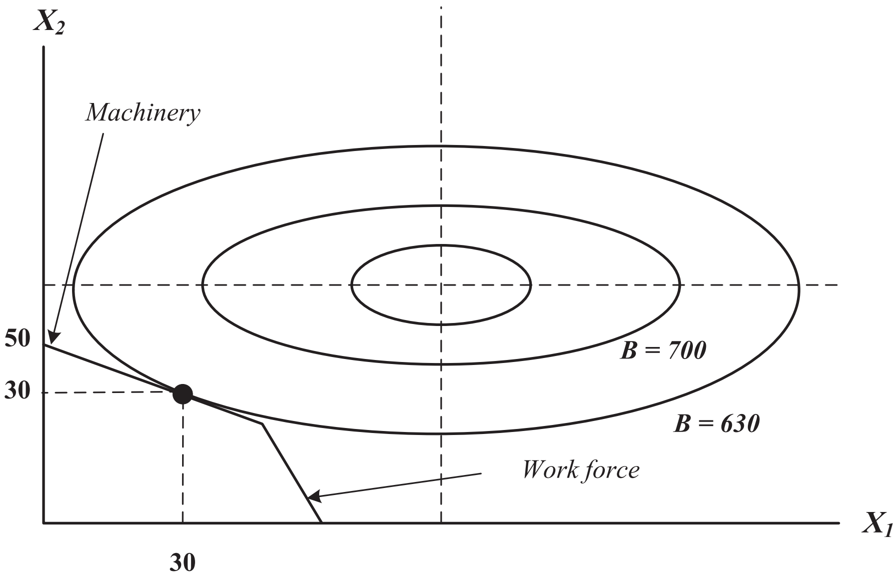


FIGURE 8.4. Nonlinear objective function

8.3 SOME APPLICATIONS

8.3.1. DETERMINATION OF TIME INTERVALS BETWEEN MACHINE ADJUSTMENTS

A machine produces parts at a constant rate of 120 units per hour. Each part generates a net income (sales price minus material cost) of 6 euros. The rate of defective parts per hour is proportional to time-period X between the adjustments of the machine, i.e., 27 X, where X=1 is one hour. Whenever the machine is adjusted it stops for 1 minute, i.e., the time required to manufacture two parts. The defective parts have to be mended in a different machine with a cost of 4 euros per part.

Profit per hour is equal to the number of parts manufactured by net profit per part minus the cost of mending the defective parts.

$$B = f(X) = (120 - 2/X)\ 6 - (27X)\ 4 = 720 - (12/X) - 108\ X$$

The objective is to determine the time-length interval between adjustments to maximize profit. If we consider that there is at least one adjustment per hour, variable X is restricted to interval [0, 1].

The optimization problem is as follows:

$$\text{Maximize } f(X) = 720 - (12/X) - 108\ X$$

$$\text{Where } 0 \leq X \leq 1$$

This model can be solved with LINGO. Below you will find the model and the solution obtained with LINGO software, which allows one to solve nonlinear programming models, in addition to linear and integer programming models.

```
MODEL:
! Determination of time interval between machine adjustments;
[PROFIT] max = 720 - (12/X) - 108*X;
X ≥ 0;
X ≤ 1;
END
```

```
Local optimal solution found at step:              8
 Objective value:                            648.0000

              Variable           Value        Reduced Cost
                     X       0.3333332           0.0000000

                   Row    Slack or Surplus      Dual Price
            BENEFICIO         648.0000            1.000000
                     2       0.3333332           0.0000000
                     3       0.6666668           0.0000000
```

8.3.2. PRODUCTION PLANNING

A company produces three products. The volume of sales of each product depends on its price, and in the case of product 3, the volume of sales also depends on the price of another product.

The marketing department estimates the following relationship between monthly sales turnover X_j (thousands of units) and unit price P_j per product:

$X_1 = 10 - P_1$

$X_2 = 16 - P_2$

$X_3 = 6 - 1/2\ P_3 + 1/4\ P_2$

The variable cost for each product is 6, 7, and 10 euros/unit respectively.

Production is limited by the machine's available time and human resources. Each month the firm has 1000 machine hours and 2000 worker hours. The consumption of these resources is shown in the following table.

Table 8.1. Resource requirements

Resource	Products		
	1	2	3
h-machine/unit	0.4	0.2	0.1
h-worker/unit	0.2	0.4	0.1

The firm wants to determine the monthly sales programming that maximizes profits, considering the resources available.

Gross profit per product is the income minus variable cost. For product 1 income is $I_1 = P_1\ X_1$ and since $X_1 = 10 - P_1$, working out the value of P_1 we get:

$$I_1 = (10 - X_1)\ X_1 = 10\ X_1 - X_1^2$$

The variable cost of product 1 is $C_1 = 6X_1$. The profit obtained is:

$$B_1 = I_1 - C_1 = 10\ X_1 - X_1^2 - 6X_1 = 4X_1 - X_1^2$$

Similarly for product 2:

$$B_2 = I_2 - C_2 = 9X_2 - X_2^2$$

The sales of product 3 depend on its price P_3 and also on P_2.

$I_3 = P_3 X_3$

$X_3 = 6 - 1/2\ P_3 + 1/4\ P_2 \rightarrow P_3 = 2(6 - X_3 + 1/4\ P_2)$ and

$P_2 = 16 - X_2$ therefore

$I_3 = 20\ X_3 - 2\ X_3^2 - 1/2\ X_3\ X_2$ and

$B_3 = I_3 - C_3 = 10\ X_3 - 2\ X_3^2 - 1/2\ X_2\ X_3$

Total profit will be

$B = B_1 + B_2 + B_3 =$

$= 4\ X_1 - X_1^2 + 9\ X_2 - X_2^2 + 10\ X_3 - 2\ X_3^2 - 1/2\ X_2\ X_3$

Therefore, the nonlinear problem to solve will be:

$$\text{Max } 4\ X_1 - X_1^2 + 9\ X_2 - X_2^2 + 10\ X_3 - 2\ X_3^2 - 1/2\ X_2\ X_3$$

$X_1, X_2, X_3 \geq 0$ (nonnegativity constraints)

$4\ X_1 + 2\ X_2 + X_3 \leq 10$ (hours-machine)

$2\ X_1 + 4\ X_2 + X_3 \leq 20$ (hours-worker)

Another situation in which we may find nonlinear production planning problems is when marginal costs vary with the production level. The constraints can also be nonlinear, when the consumption of one resource is not strictly proportional to the amount produced.

The model and solution obtained with LINGO are the following:

```
MODEL:
! Production planning;
[PROFIT] max = 4*X1-X1^2+9*X2-X2^2+10*X3-2*X3^2-0.5*X2*X3;
[MACHINE] 4*X1 + 2*X2 + X3 < 10;
[WORKERS] 2*X1 + 4*X2 + X3 < 20;
END
```

```
Local optimal solution found at step:            10
 Objective value:                              28.82051

                Variable           Value      Reduced Cost
                      X1       0.4102567         0.0000000
                      X2        3.230769      -0.1716926E-07
                      X3        1.897436      -0.3105690E-08


                     Row     Slack or Surplus       Dual Price
                 PROFIT          28.82051             1.000000
                 MACHINE         0.0000000           -0.7948716
                 WORKERS         4.358976            0.0000000
```

8.3.3. SOME CHARACTERISTICS OF NONLINEAR OPTIMIZATION METHODS

Solving problems such as those in previous section, where the maximum or minimum of a nonlinear function should be found, is based on approximation function methods. At each iteration, these algorithms find the optimal solution of X by approximating the function through a simpler function, for example, quadratic at each iteration. The optimum of the approximating function is found analytically. By repeating the procedure the methods will converge in an optimum to the original function. The different methods differ with respect to the data required and the type of approximation performed. Thus, Newton's method uses the first and second derivatives at a point to generate a quadratic approximation. Figure 8.5 shows Newton's method for the problem of section 8.3.1.

MAX $f(X) = 720 - (12/X) - 108\,X$

Where $0 \le X \le 1$

Newton's method rapidly converges to the optimal if the starting point is close to it. Unfortunately, it does not always converge. It may diverge or shift without converging or it may find a minimum rather than a maximum.

Another weak point of this algorithm is that it requires us to know the second derivative of the function. It may happen that the second derivatives do not exist or are costly to evaluate. "Quasi" Newton methods overcome this drawback by estimating the values of the second derivatives.

Many nonlinear optimization techniques follow this algorithmic structure:

1. The algorithm is initiated with a solution $\mathbf{X}^1 = (X^1{}_1, X^1{}_2, ..., X^1{}_n)$

2. Find a movement direction away from current solution that improves the objective function value $f(\mathbf{X}) = f(X_1, X_2 ... X_n)$

3. Determine how far the current solution should shift in the direction of the improved objective function, in other words, find the size of the modification.

4. Repeat steps 2 and 3, always using the last solution obtained in 3, until no other improvement direction of the objective function can be found or until improvement is lower than a fixed quantity. The optimal solution is the last solution obtained in this procedure.

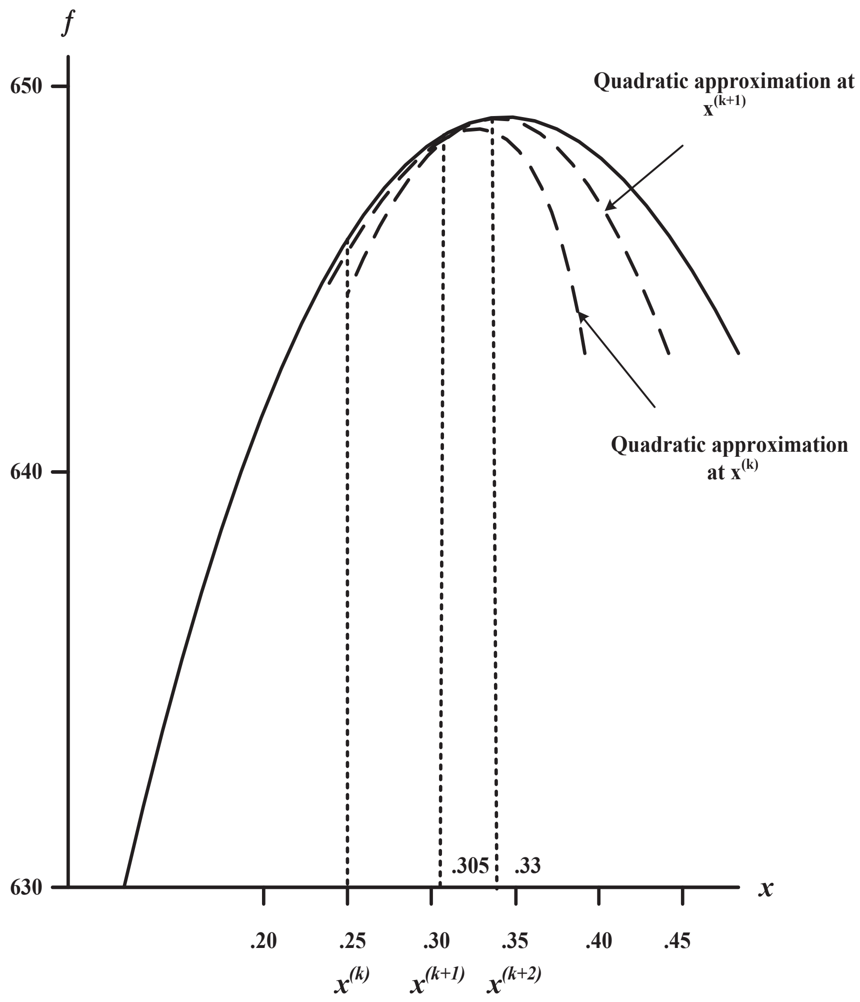


Figure 8.5. Newton's method for maximizing f(X)

Most techniques based on this structure differ in the method of developing steps 2 and 3. In particular, the gradient search method selects the gradient vector in a solution **X** as the moving direction at step 2 of the general algorithmic structure. The gradient vector of f, ∇f is the vector whose components are the partial derivatives of f with respect to the Xi: $\nabla f = (\partial f/\partial X_1, \partial f/\partial X_2, ...\partial f/\partial X_n)$.$\nabla f(\mathbf{X})$ is a vector that indicates the direction in which f values change more rapidly from point **X**. Therefore, it is locally the best moving direction.

An analogy can illustrate the gradient search procedure. Suppose that a shortsighted person wants to climb to the top of a hill, so he cannot see the top of the hill in order to walk directly in that direction. However, when he stands still he can see the ground around his feet well enough to determine the direction in which the hill is sloping upward most sharply. Then he starts walking in that fixed direction and continue as long as he is still climbing. He eventually stops at a new trial location when the hill becomes level in this direction, at that point he prepares to do another iteration in another direction. He continues this procedure, following a zigzag path iteration up the hill, until he reaches a trial location where the slope is essentially zero in all directions. Under the assumption that the hill is concave, he must be at the top of the hill.

There is no one method which is the best to solve the majority of problems. This is the main challenge for nonlinear programming. In general, the algorithms are appropriate for solving particular types of problems. In some cases, it is possible to solve nonlinear programming problems approximating them through linear programming models. This is the essence of separable programming. Another case of particular relevance is when the constraints are linear and the objective function is a second order function: quadratic programming. In the next section we deal with a real application, such as portfolio selection.

This brief and intuitive explanation of nonlinear programming methods helps us to understand the options which LINGO and *Solver* in Excel have and, especially, to use them better in order to improve decision making. See options in the annexes 1 and 2. In particular LINGO uses sequential linear programming and the generalized reduced gradient method. LINGO also includes an optimizer to obtain the global optimum in non-convex models (*Global Solver*). Finally, it is interesting to point out that interior point methods are based on the ideas explained in previous sections.

8.4. EFFICIENT PORTFOLIOS

8.4.1. THE MARKOWITZ MODEL

Markowitz and Sharpe won the Nobel Prize in 1990 due to their portfolio selection models. The Markowitz model permits the generation of optimum portfolios, which is the basis of modern theory of portfolios and is opening a new phase in financial analysis.

The problem is as follows: The asset return is a random variable which can be characterized by its mean and variance. Investors are usually interested in total return (expected value) as well as in the associated risk of their investments. This is an example of a problem with two conflicting objectives. The investor wants the maximum return and the minimum risk. In addition, the objective function is nonlinear. Therefore, portfolio managers use nonlinear programming models as decision making tools to determine the investment portfolio with the best combination of total return and risk. This is the portfolio that best fits the preferences and characteristics of investors.

According to Markowitz the expected rate of return of a portfolio is found by taking the weighted sum of the individual expected rates of return from the n assets that compose it, while the risk of the portfolio depends on three components: the assets proportion of the portfolio, the variance of asset returns and the covariance of the returns for each pair of assets.

The sources that contribute to the return variability of stock assets (price and dividend) are risk elements. Risk can be divided in two parts: systematic and non-systematic risk. The former depends on causes external to the company (political changes, inflation, interest, etc.), while the latter – non-systematic risk - depends only on factors internal to the enterprise (leadership, indebtedness, markets in which it competes, etc.).

Diversification also allows us to reduce the risk of the portfolio. Naive diversification is based on combining values from different sectors, whereas scientific diversification of Markowitz model is based on combining values having a correlation or covariance that allows the reduction of risk without sacrificing profitability. In general, the smaller the correlation between the values of the portfolio, measured by the covariance, the lower the its risk. In short, the essence of the scientific diversification is to combine values whose covariance is negative.

As non-systematic risk is unique and particular to each company, the investor can avoid this type of risk by diversifying and investing in stocks of companies whose covariance or correlation is adequate. Several studies have shown that the diversification of risk permits the avoidance of non-systematic risk and that it is achieved by including 12-15 values in the portfolio, no improving the results if we increase this value.

In short, **efficient portfolios** are those from which the non-systematic risk has been removed and the risk is defined exclusively by the movements of the market. In practice there are holding companies, investment and pension funds, etc. who invest their funds in a wide range of values, which are well above those considered necessary. Therefore, the profitability of their portfolios is lower due to the level of risk they take, primarily for two reasons: excessive increased management costs and resorting to values with low returns, having many types of securities in its portfolio. Therefore, the portfolios formed by the scientific method of Markowitz dominate the feasible region of portfolios formed by traditional methods.

Firstly, we shall see the general formulation of the Markowitz model for portfolios selection and secondly, its application to an example by using Excel spreadsheets and the *Solver* tool to solve the resulting nonlinear programming model.

Suppose that we want to form a portfolio from n assets. The decision **variables x_i (i=1, 2…n)** represent the proportion of each asset in the portfolio.

And **r_i is the average return of the asset i** and **σ_{ii}^2** the estimated **variance** of return on asset i. We measure asset risk by this variance. **σ_{ij}^2** for i, j=1, 2,…n (i≠j) is the **covariance of returns of assets i and j**.

The expected rate of return of a portfolio is the weighted sum of the individual average returns from the n assets that compose it, where the weights are the proportion of each asset and represent the decision variables of the model

$$R(X) = \sum_{i=1}^{n} r_i x_i$$

And the **variance of portfolio total return *V(X)*** is as follows.

$$V(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij}^{2} x_i x_j$$

In matricial form:

$$V\ (X) = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \begin{bmatrix} \sigma_{11}^2 & \cdots & \sigma_{1n}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{n1}^2 & \cdots & \sigma_{nn}^2 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

The portfolio selection model can be formulated as follows: Find the values of the decision variables **xi (i = 1, 2... n)** representing the proportion of each value in the portfolio that minimises the total risk.

$$MIN \quad V(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij}^{2} x_i x_j$$

As constraints of the model we include one in order to obtain a minimum return and another that represents the sum of proportions of each asset that should be 1:

$$\sum_{i=1}^{n} r_i x_i \geq R_{MINIMUMPORTFOLIO}$$

$$\sum_{i=1}^{n} x_i = 1$$

And also nonnegativity constraints of the variables:

$$x_i \geq 0 \quad for \quad i = 1,2,...n$$

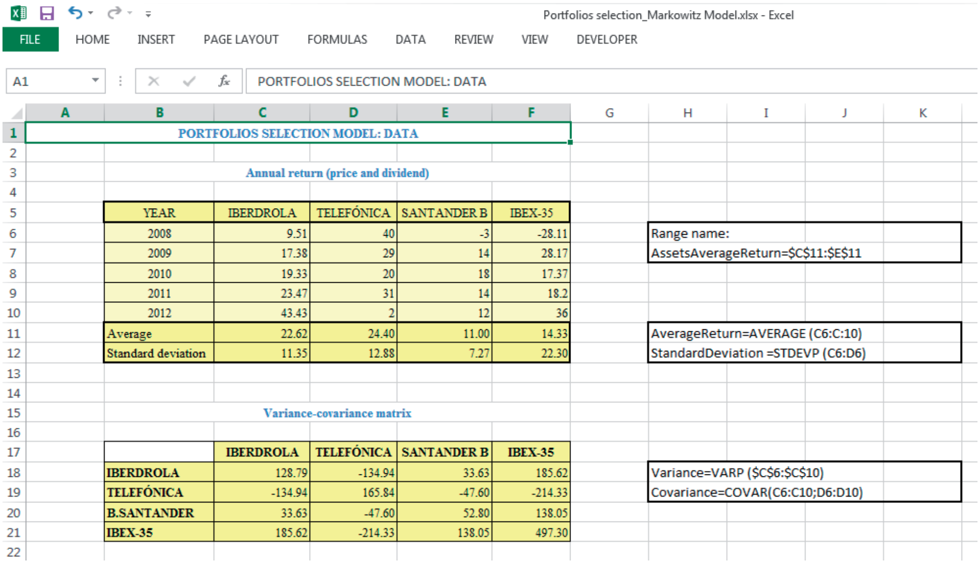
It is also usual to consider the upper and lower bounds of the variables. In a multiobjective problem you obtain efficient solutions by optimising an objective (minimizing the risk of the portfolio) and considering another objective as a parametric constraint (return on the portfolio). By modifying the minimum return on the portfolio and solving the resulting quadratic programming model (the objective function is second grade nonlinear) we obtain efficient portfolios, which are those in which we cannot improve one objective without worsening the other. The concept of efficient solution is explained in chapter 6.

We can see the practical application of this model in Excel, since this tool allows us to easily calculate all the coefficients, solve the nonlinear model with *Solver*, and draw the efficient frontier which represents the combinations of return and risk, measured by the standard deviation of the portfolio. The efficient frontier permits us to decide on the best composition of portfolio taking into account the preferences of investors.

Figure 8.6 shows the annual return of three types of securities between 2008 and 2012. The data are not real and for pedagogical reasons the data series is small, as well as the number of assets that can be part of the portfolio. In practice we take monthly series to calculate the covariance. Instead of only using historical data current data can be used on the expected return of an asset. We have calculated average returns for securities, their standard deviation and variance-covariance matrix in Figure 8.6. These data are used in the portfolios model presented in Figure 8.7.

"*Make Unconstrained Variables Non-Negative*" should be selected to solve the model with all nonnegative variables. Otherwise all of these constraints should be introduced.

If we change the desired minimum return on the portfolio and solve the resulting model, we can obtain the composition and the risk associated with the optimal portfolio for a given total return. In this way it can generate the efficient frontier in which we have the minimum risk for each portfolio total return, as we can see in figure. 8.9. This model can also be formulated using the LINGO modelling language. The LINGO programming capabilities allow us to solve a series of models and draw the efficient frontier.



Portfolios selection_Markowitz Model.xlsx - Excel

A1 PORTFOLIOS SELECTION MODEL: DATA

PORTFOLIOS SELECTION MODEL: DATA

Annual return (price and dividend)

YEAR	IBERDROLA	TELEFÓNICA	SANTANDER B	IBEX-35
2008	9.51	40	-3	-28.11
2009	17.38	29	14	28.17
2010	19.33	20	18	17.37
2011	23.47	31	14	18.2
2012	43.43	2	12	36
Average	22.62	24.40	11.00	14.33
Standard deviation	11.35	12.88	7.27	22.30

Range name:
AssetsAverageReturn=$C$11:$E$11

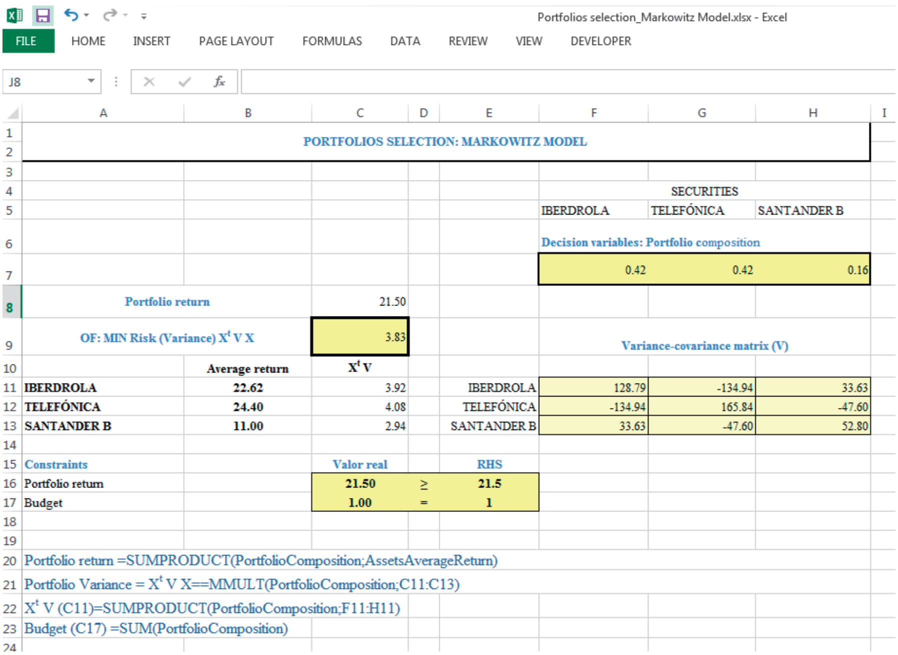
AverageReturn=AVERAGE (C6:C:10)
StandardDeviation =STDEVP (C6:D6)

Variance-covariance matrix

	IBERDROLA	TELEFÓNICA	SANTANDER B	IBEX-35
IBERDROLA	128.79	-134.94	33.63	185.62
TELEFÓNICA	-134.94	165.84	-47.60	-214.33
B.SANTANDER	33.63	-47.60	52.80	138.05
IBEX-35	185.62	-214.33	138.05	497.30

Variance=VARP ($C$6:$C$10)
Covariance=COVAR(C6:C10;D6:D10)

Figure 8.6. Returns and variance-covariance matrix



Portfolios selection_Markowitz Model.xlsx - Excel

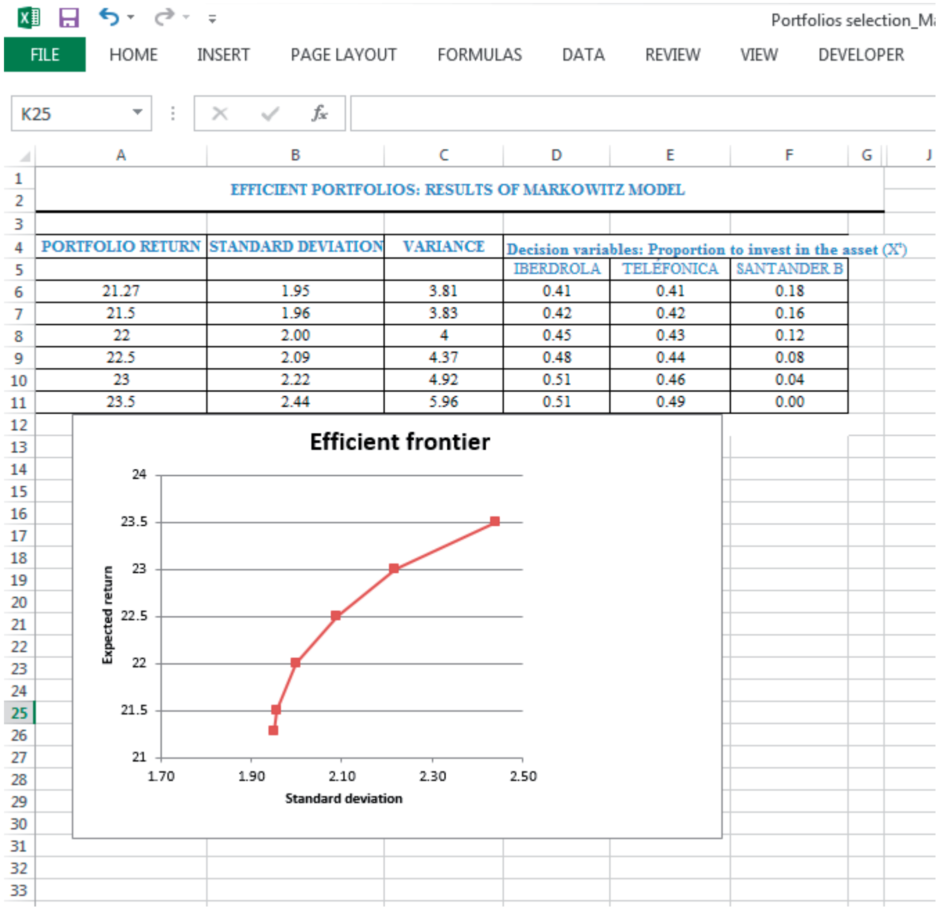
J8

PORTFOLIOS SELECTION: MARKOWITZ MODEL

					SECURITIES	
				IBERDROLA	TELEFÓNICA	SANTANDER B
				Decision variables: Portfolio composition		
				0.42	0.42	0.16
Portfolio return		21.50				
OF: MIN Risk (Variance) X^t V X		3.83			Variance-covariance matrix (V)	
	Average return	X^t V				
IBERDROLA	22.62	3.92	IBERDROLA	128.79	-134.94	33.63
TELEFÓNICA	24.40	4.08	TELEFÓNICA	-134.94	165.84	-47.60
SANTANDER B	11.00	2.94	SANTANDER B	33.63	-47.60	52.80
Constraints		Valor real	RHS			
Portfolio return		21.50 ≥	21.5			
Budget		1.00 =	1			

Portfolio return =SUMPRODUCT(PortfolioComposition;AssetsAverageReturn)
Portfolio Variance = X^t V X==MMULT(PortfolioComposition;C11:C13)
X^t V (C11)=SUMPRODUCT(PortfolioComposition;F11:H11)
Budget (C17) =SUM(PortfolioComposition)

Figure 8.7. Markowitz model to obtain efficient portfolios



PORTFOLIO RETURN	STANDARD DEVIATION	VARIANCE	Decision variables: Proportion to invest in the asset (X^i)		
			IBERDROLA	TELÉFONICA	SANTANDER B
21.27	1.95	3.81	0.41	0.41	0.18
21.5	1.96	3.83	0.42	0.42	0.16
22	2.00	4	0.45	0.43	0.12
22.5	2.09	4.37	0.48	0.44	0.08
23	2.22	4.92	0.51	0.46	0.04
23.5	2.44	5.96	0.51	0.49	0.00

Figure 8.8. Efficient frontier

8.4.2 SHARPE MODEL

A drawback of the Markowitz model is the large amount of data, in particular the number of covariances, whose value depends on the number of securities (n) that is $(n^2-n)/2$. Thus, if we consider 10 securities for the portfolio, we need 45 covariances, 1225 in case of n=50, 4950 when n=100 and nearly half a million data when n=1000.

The Sharpe model, among others, has tried to simplify the process of obtaining the data required for the application of the Markowitz model. Basically, the Sharpe model requires n data that is equal to the number of securities considered in the portfolio. This model introduces a coefficient beta, which measures the sensitivity of the securities return when compared to market movements. It shows us both the risk of an asset and the effect on the risk of the portfolio which includes this asset. This beta coefficient is obtained by comparing the return of each asset with an economic or stock market index by applying linear regression.

Sharpe replaced the covariance of each pair of assets with this coefficient beta of the relationship between each asset and the market in general, defining the expected return on securities as a function of the market profitability and representing it by the following regression equation:

$$r_{it} = \alpha_i + \beta_i\, r_{It} + e_{it}$$

where:

r_{it} = return of asset i in period t
α_i = alfa coefficient, the point of intersection of the regression line with the axis line
β_i = beta coefficient, slope of regression line
r_{It} = return of market index I in period t
e_{it} = random error of regression where $E(e_{it}) = 0$ y $Var(e_{it}) = \sigma_i^2$

Alpha coefficient is the point of intersection of the regression line with the axis and indicates the expected return on i asset when the market return is zero. It is advisable to perform the regression on the excess of return, that is, once a constant value for the price of money without risk is deducted.

Beta coefficient: This is the most important coefficient. Beta is the slope of the regression line and measures the sensitivity of the i asset return with respect to the movements of the market index. If Beta = 1, it indicates a variability of identical return for the asset and index. If Beta > 1, it indicates a higher variability for the asset than the index and if Beta < 1, it indicates a lower variability of the asset return than the index.

As we have seen, the systematic risk is the variability in the price of an asset due to fluctuations in the market. Therefore, we can measure the systematic risk using the beta coefficient. Beta not only shows us the degree of response in the asset price in relation to the movements of the market index, but it also quantifies the response to other assets.

The random error of an asset is the non-systematic risk of that asset. This risk can be described by the scatter about the regression line. The greater the dispersion of the point cloud the bigger the non-systematic risk of a value.

The non-systematic risk can be measured by the residual variance e_{it}^2 by using the following formula:

Non-Systematic Risk = e_{it}^2 = *Total Risk – Systematic Risk*

The beta of a portfolio is simply the weighted average of the betas of the assets, which simplifies the risk calculation of the portfolio a lot, as we do not estimate the covariance between each pair of values. If we change the desired minimum return on the portfolio and solve the resulting model, we can obtain the composition and the risk associated with the optimal portfolio for a given total return. In this way it can generate the efficient frontier in which we have the minimum risk for each portfolio total return, as we can see in figure. 8.9. This model can also be formulated using the LINGO modelling language. The LINGO programming capabilities allow us to solve a series of models and draw the efficient frontier.

If x_i is the asset weight in the portfolio, the portfolio variance taking into account that $r_{it} = \alpha_i + \beta_i r_{It} + e_{it}$ is as follows:

$$V(X) = Var\left[\sum_{i=1}^{n} x_i \left(\alpha_i + \beta_i r_I + e_i \right) \right]$$

$$V(X) = \left(\sum_{i=1}^{n} x_i \beta_i \right)^2 e_I^2 + \sum_{i=1}^{n} x_i^2 e_i^2$$

Thus, Sharpe´s model can be written:

$$MIN \left(\sum_{i=1}^{n} x_i \beta_i \right)^2 e_I^2 + \sum_{i=1}^{n} x_i^2 e_i^2$$

with the constraints:

$$\sum_{i=1}^{n} x_i = 1$$

$$\sum_{i=1}^{n} x_i \left(\alpha_i + \beta_i r_I \right) \geq R_{MINIMUM\ PORTFOLIO}$$

Or

$$MIN \quad Z^2 e_I^2 + \sum_{i=1}^{n} x_i^2 e_i^2$$

with the constraints:

$$Z - \sum_{i=1}^{n} x_i \beta_i = 0$$

$$\sum_{i=1}^{n} x_i = 1$$

$$\sum_{i=1}^{n} x_i \left(\alpha_i + \beta_i r_I \right) \geq R_{MINIMUM\ PORTFOLIO}$$

Let us solve this model by using Excel Solver with the same data that we used to solve the Markowitz model. Figure 8.9 shows data from the annual returns of the three assets from 2008 until 2012, as well as the annual returns of the IBEX-35 which is the index of the market that we have used. Based on this information, we estimate the data required to apply the Sharpe's model of, presented in Figure 8.10 along with range names and the most important formulas for the calculations.

PORTFOLIOS SELECTION MODEL: DATA - SHARPE MODEL

Annual return (price and dividend)

YEAR	IBERDROLA	TELEFÓNICA	SANTANDER B	IBEX-35
2008	9.51	40.00	-3.00	-28.11
2009	17.38	29.00	14.00	28.17
2010	19.33	20.00	18.00	17.37
2011	23.47	31.00	14.00	18.20
2012	43.43	2.00	12.00	36.00
Average	22.62	24.40	11.00	14.33
Standard deviation	11.35	12.88	7.27	22.30

Residuals.Iber	ResidualsTel.	ResidualsSan
2.73	-2.69	-2.22
-10.41	10.57	-0.84
-4.43	-3.09	6.16
-0.60	8.27	1.92
12.72	-13.06	-5.02
0.00	0.00	0.00
7.71	8.57	3.81

Sharpe model

	ALFA	BETA	R2	Residuals Variance
IBERDROLA	17.28	0.37	0.54	59.50
TELEFÓNICA	30.57	-0.43	0.56	73.47
SANTANDER B	7.02	0.28	0.73	14.48
IBEX-35				497.30

Decision variables: Portfolio composition

IBERDROLA	0.00
TELEFÓNICA	0.00
SANTANDER B	0.00

Portfolio return	0.00
Portfolio variance	0.00

Portfolio return=J18*C11+J19*D11+J20*E11

Portfolio variance = (SUMPRODUCT(PortfolioComposition;BetaValues))^2*ResidualsVarianceIBEX35+SUMPRODUCT(PortfolioComposition^2;ResidualsVariance)

Figure 8.9. Data: Returns and beta coefficients of Sharpe

PORTFOLIOS SELECTION:SHARPE MODEL

SECURITIES

IBERDROLA	TELEFÓNICA	SANTANDER B
Decision variables: Proportion to invest in the asset		
0.44	0.44	0.12

Portfolio return	22.00
OF: MIN Risk (Variance)	25.94

	Average return	Portfolio composition	ALFA	BETA	ResidualsVariance
IBERDROLA	22.62	0.44	17.28	0.37	59.50
TELEFÓNICA	24.40	0.44	30.57	-0.43	73.47
SANTANDER B	11.00	0.12	7.02	0.28	14.48
IBEX-35					497.30

Constraints	Value		RHS
Portfolio return	22.00	≥	22.00
Budget	1.00	=	1.00

Portfolio return =SUMPRODUCT(C11:C13;B11:B13)

Portfolio variance=(SUMPRODUCT(PortfolioComposition;Beta))^2*ResidualsVarianceIBEX35+SUMPRODUCT(PortfolioComposition^2;ResidualsVariance)

Figure 8.10. Efficient portfolio model of Sharpe

In both Markowitz and Sharpe models, it is common to add constraints in order to limit the value of the decision variables so that neither of them represents a very large portfolio fraction. Finally, we can say that there are many other possible models and you can find finance treatises about them.

8.5. SUMMARY

The most important feature of nonlinear programming is that no one method is "the best" to solve any nonlinear model. The optimal solution is not a corner point of the feasible region and it can even be an interior point. When function objective and/or constraints are nonlinear, the procedure to find the optimal solution is more complicated than in the linear case. However, there are methods that solve certain types of problems efficiently and many applications in business administration and management that need these methods to improve decision-making. An especially important one is the efficient portfolios of securities investment. Markowitz and Sharpe models are the basis of the modern portfolio theory in finance.

8.6. SELECTED REFERENCES

1. Daellenbach, H.G.; George, J.A.y D.C.McNickle (1987): *Introducción a las técnicas de investigación de operaciones*. CECSA.

2. Hillier, F.S. and Lieberman G.J. (2010): Introduction to Operations Research. Ninth Edition. McGraw-Hill.

3. Hillier, F.S and Hillier, M.S. (2011): Introduction to Management Science. A Modelling and Case Studies Approach with Spreadsheets. Fourth Edition. McGraw-Hill.

4. LINDO Systems (2011): *LINGO. The modeling language and optimizer*.

5. LINDO Systems (2006): *Optimization in Spreadsheets with What´ s Best!* http://www.lindo.com.

6. Luenberger, D.G. (1998): Investment Science. Oxford University Press. Chapters 6 and 7.

7. Winston, W. L. and S.C. Albright (2012): Practical Management Science. Fourth Edition. South-Western. USA.

8.7. CASE STUDIES

CASE STUDY 1

Solve the following model which represents a problem of investment portfolio graphically. Analyse the differences between this case and graphical solution of a linear programming model.

Min $0.09\ X_1^2 + 0.04\ X_1\ X_2 + 0.06\ X_2^2$

Constraints:

$$X_1 + X_2 = 1$$

$$0.06\ X_1 + 0.02\ X_2 \geq 0.03$$

$$X_1 \leq 0.75$$

$$X_2 \leq 0.9$$

$$X_1 \geq 0$$

$$X_2 \geq 0$$

CASE STUDY 2

Suppose that table 8.2 includes the historical returns of five types of assets and the IBEX-35.

Table 8.2. Annual returns

YEAR	ASSET 1	ASSET 2	ASSET 3	ASSET 4	ASSET 5	IBEX
1999	12,5	19,9	4,1	12,1	5,6	12,0
2000	9,4	9,2	2,6	8,3	6,2	8,3
2001	13,9	19,9	-4,2	10,2	4,5	9,4
2002	7,7	14,3	7,9	8,4	11,2	10,5
2003	8,7	19,2	9,9	8,6	0,9	9,8
2004	11,0	15,8	10,1	6,9	8,6	11,2
2005	8,8	18,2	6,3	6,5	8,5	8,5
2006	10,5	18,6	10,0	7,3	11,6	10,9
2007	12,5	14,5	2,2	8,4	12,9	10,1
2008	14,0	18,2	6,8	7,2	8,7	12,0
2009	6,8	13,6	8,5	10,3	8,8	8,9
2010	14,2	17,1	6,1	12,8	8,4	14,3
2011	10,9	10,0	6,0	11,3	4,3	8,4
2012	9,5	8,5	6,0	7,9	7,3	9,3

Obtain the composition of efficient portfolios for three types of investors, high, medium and low return by using Markowitz and Sharpe models. This problem can be solved with Excel Solver and LINGO.

CHAPTER 9

METAHEURISTIC TECHNIQUES: GENETIC ALGORITHMS

In recent decades a new group of algorithms has appeared, called metaheuristics, and they have been successfully applied to a variety of difficult search and combinatorial optimization problems. They are the latest generation of heuristic algorithms, widely used for solving optimization problems of all kinds when exact methods are not applicable. A metaheuristic algorithm can be defined as an iterative process that guides and/or modifies the operations and/or solutions of one or more subordinate heuristic algorithms to produce higher quality solutions in a reasonable time (Voss et al., 1999). They manipulate a single solution or a combination of these in each iteration, and as the process proceeds the solution or solutions improve.

Among the metaheuristics that have appeared we may include those that mimic the behavior of natural systems, both biological and physical, such as the natural evolution of the species, thermodynamics, cooperative work in an ant colony or the behavior of neurons in the brain. From the wide variety of existing metaheuristics, we can highlight, due to their excellent results and the diversity of problems to which they are being applied; genetic algorithms, tabu-search and simulated annealing, to which we will devote this chapter. We can find a detailed description of these and other metaheuristics in Rayward-Smith et al. (1996). Among their applications we can cite routing problems, schedule management, production scheduling, project scheduling with limited resources and many others. Genetic algorithms are explained in detail in the next section, and in the last two points of this chapter tabu-search and simulated annealing are outlined.

9.1. GENETIC ALGORITHMS

In the '60s the rapid proliferation of computers led to their use as simulation tools by the scientific community. In the early '70s, a group of researchers from the University of Michigan, led by Professor John Holland (1975), proposed genetic algorithms as computer programs that mimicked the natural evolutionary process and behaved robustly in a variable and uncertain environment. The main theme of the research focused on the robustness of such systems, i.e. how to find the right balance between efficiency and effectiveness to suit different environments. The robustness of the systems, both software and hardware, was a crucial aspect of their design, as the cost of rehabilitation and redesign could be drastically reduced or even eliminated.

The resolution of a particular complex problem can be viewed as search in a space of possible solutions and, since we generally look for the best solution, it can be further understood as an optimization problem. Genetic algorithms are algorithms whose search mechanisms imitate a natural phenomenon: the evolution of species through genetic inheritance. In nature, the problem that each species faces is seeking improvements for its own adaptation to the environment. The main idea of genetic algorithms is to do what nature does.

In nature, the members of a population compete with each other for resources such as food, water or shelter. At the same time, males compete amongst themselves to attract females. Individuals who are better prepared for survival and who attract females will be those that get more offspring. The less successful individuals will have fewer offspring or even none. This means that the genes of the fittest will be inherited by a growing number of individuals in successive generations. The combination of the good features of past generations usually produces individuals that are better adapted than their predecessors. In this sense the species evolves into a kind that is better and better adapted to their environment.

Let us take a population of rabbits as an example (Michalewicz, 1996). In this population some rabbits are faster and cleverer than the rest. The foxes that feed on these rabbits will find it more difficult to catch fast and clever rabbits, so those will surely survive and do what rabbits do best: make more rabbits. Of course, some of the slow and clumsy rabbits survive due to good luck and may possibly have offspring. The offspring population becomes a good mix of genetic material: fast rabbits have crossed with slower rabbits, some faster rabbits with other very fast ones, slow rabbits with clever ones, clumsy with fast, etc. Additionally, mutations in the genetic material of some individuals in the population can be produced, which may introduce different characteristics from those inherited from previous generations, i.e. introduces greater variability in the population. As a consequence, the resulting rabbit population will be, on average, faster and more “intelligent" than the original population, because most of the parents which survived the foxes were quick and clever. It is interesting to think that the fox population also suffers a similar evolution, since otherwise, the rabbits would become too fast and intelligent to be caught by them.

Genetic algorithms follow nature procedure in the previous example step by step. They work on a population of individuals, each of which represents a possible solution to the problem they are applied to. Each individual is assigned a fitness value representing the quality of the solution. The individuals in the population cross with each other to produce new solutions, so that individuals with a better fitness value are more likely to be selected for crossover. When two individuals or solutions are selected for crossover, they produce one or more solutions (children) who inherit some of the characteristics of each of the parents. The least qualified individuals, i.e. the solutions with worse fitness value are less likely, but they also have some possibility to cross with other solutions and pass their features on to the next generation.

Figure 9.1 shows, in pseudo code, the general procedure of a genetic algorithm. The first step consists of generating the initial population of solutions. One of the main differences between the genetic algorithms and other sequential algorithms, such as tabu search or simulated annealing, is that the first one manages a set of solutions in every iteration and not only one solution as it is the case of the last two. Once we have the initial population that may have been obtained in a random way, each individual is evaluated, that is, a fitness value is assigned to every individual.

```
Genetic Algorithm Procedure

   Create_initial_population
   Evaluate_population
   While not (termination_condition)
      Select_population
      Cross_population
      Mutate_population
      Evaluate_population
```

Figure 9.1. Genetic Algorithm: general procedure

After all of the members of the initial population have been evaluated, the following steps are repeated until the stopping condition is satisfied. Firstly, the selection of the population is carried out. In this process each of the individuals in the population is copied a number of times, so that the best individuals generally have a higher number of copies than the less skilled individuals. In this way we obtain a new population that replaces the previous one.

After that, the crossover process is carried out. The individuals in the population are paired at random and every couple undergoes crossover with a given probability. If the operation is carried out by the couple (parents), two new solutions (children) that replace the previous ones are created. If not, the parents remain unaltered. Thus, in the resulting population individuals from different generations can live together. The effect of the selection process which is carried out before the crossover operation is that the best individuals in the population participate more actively in such processes.

Finally, some individuals in the population can mutate, i.e. some solutions can be partially altered, allowing the population to introduce new features or material which have been lost through evolution.

The resulting population then is re-evaluated and the termination condition is checked. This condition generally refers to the elapsed computation time, the number of generations or iterations performed, the number of individuals who have been evaluated, the improvement produced in the last iterations, the variability of the population, etc.

Obviously, we need to design an encoding for the solutions before we can start applying a genetic algorithm, this being one of the fundamental aspects of the design and therefore the subsequent efficiency of the genetic algorithm. That is, we need to define how to represent each of the possible solutions in an appropriate way.

9.1.1. SOLUTIONS ENCODING

The genetic algorithm operates on an encoded representation of the solutions, equivalent to the genetic material of the individuals, rather than directly on the given solutions. A solution to the problem can be represented as a set of parameters. These parameters, known as genes, can be placed one after the other, forming a chain of values, which are referred to as a chromosome. In genetic terms, the parameter set represented by a chromosome is called a genotype. It contains the information necessary to construct an organism, known as a phenotype.

In the genetic algorithm proposed by Holland each of the solutions is represented by a chain on a binary alphabet, i.e. a chain consisting of only zeros and ones. Although the simple genetic algorithm (proposed by Holland) used a binary encoding, other types of encodings have been developed, such as strings of integers or real numbers and even chains in which genes do not contain numbers. That means that the strings can be defined using any alphabet.

Sometimes the chromosome does not directly represent a solution to the problem, but the information needed by a particular algorithm to solve the specific problem, such as for example, a heuristic algorithm which is able to find good solutions to the problem. In this case, we should apply such a procedure using the information represented in the chromosome in order to obtain a solution. Regardless of whether they directly represent a solution to the problem or not, individuals from the population are generally referred to as solutions. An appropriate encoding of the solutions is crucial to the success of the genetic algorithm, and the rest of the procedures to be designed that will manipulate and act on the solutions will depend on this encoding.

Once we have defined the encoding, the next step is to create the initial population of solutions of a certain size. There are generally two ways to generate the individuals of that population: at random or using some heuristic algorithm. The advantages of the first mechanism are the required time and the diversity of the generated solutions, but it may have the disadvantage of creating a mediocre population which may need a lot of time to converge to good solutions. The second method does not have the above mentioned drawback, since the solutions are usually of a better quality, so it can converge faster. However, we should **avoid** generating a **population with little diversity**, because this lack of diversity could lead to a **premature convergence**, by being trapped in a **local optimum**. We must also take into account the time needed to create the population by this method because, if it is too high, a random population may be advisable.

9.1.2. FITNESS FUNCTION

The fitness function or evaluation function is the one that assigns, to each of the individuals in the population, a fitness value which indicates the suitability of that individual with respect to other individuals who are part of the population.

In some problems it is easy to assess the quality of a given solution with a single value. For example, if we are looking for the point that maximizes the value of a function then the solutions that produce a greater value will be those that have a better fitness value, and the fitness value even could match the value that is produced in the function to maximize. However, in certain problems it may not be so easy to evaluate the quality of a solution with a single number. In the task of planning and scheduling a project, we may have different aspects to optimize, such as to minimize its overall duration, to level the demand of resources, to obtain an acceptable ratio cost/duration etc., so it may not be easy to evaluate the solutions with a single fitness value.

Sometimes, once the evaluation process is finished, i.e. once the fitness value has been assigned to each of the individuals in the population, it may be recommended to make a scaling of these values to avoid premature convergence of the algorithm. The scaling technique will be discussed in the next section, after having described the selection process.

9.1.3. SELECTION

The selection process performed by genetic algorithms is an artificial version of natural selection of species, based on the principle of "survival of the fittest". That is, in nature the individuals that succeed are the best equipped and better prepared to survive the obstacles imposed by nature, such as predators, diseases, lack of resources, etc. The rest, less skilled individuals, are not able to overcome such obstacles and die.

In genetic algorithms, the adequacy and preparation of an individual is represented, as mentioned earlier, by the fitness value. Therefore, the individuals with a better fitness value are those who are more likely to remain in the population and breed with other individuals and pass their genetic material on to future generations. The individuals with a worse fitness value are more likely to disappear.

To simulate this natural process, each individual is copied a number of times to form an intermediate population which replaces the previous population and is the same size as the latter. That is, the best individuals receive more copies than the worst, who may not even receive any copy at all and will therefore disappear.

There are two important aspects of the behaviour of the genetic algorithm associated with the selection process and which are closely interrelated: population diversity and selection process pressure. If the selection process is very hard, i.e. if only highly qualified individuals survive and the rest die, then the population loses diversity and this fact can lead to premature convergence of the genetic algorithm, possibly ending trapped in a local optimum.

If the pressure of the selection process was very weak, i.e., if all individuals in the population are always able to survive, the search process would become "blind" because it would not be guided by the best individuals. It is therefore an important aspect to find a suitable compromise between the diversity of the population and the level of pressure exerted by the selection process. The selection process can be implemented in various ways. We will briefly explain some of them in the following lines.

The *simple selection* is the simplest mechanism. It consists of simulating a spinner in which each solution is assigned a section thereof. The size of the section allocated to a solution is in proportion to its fitness value, so that the probability that an individual *i* is selected, is given by:

$$\text{Pselect\_i} = \frac{\text{f(i)}}{\sum_{j=1}^{pop\_size} f(j)}$$

The best solutions will have the biggest sectors and therefore, the worst solutions will be assigned the smaller sectors, but every solution in the population will have their sector assigned at roulette. The next step is to throw the ball as many times as solutions appear in the population, i.e. pop_size times. Every time the ball is released and it falls in a given sector, the solution represented by this sector is copied to a new intermediate population, so that on average the best solutions get the largest number of copies and the worst individuals get less copies or even none. The population thus obtained replaces the previous population and these individuals will participate in the crossover process.

In *deterministic sampling*, the expected number of copies for each solution to receive is given by Ei=Pselect_i * pop_size where Pselect_i is calculated as in the previous case. Each solution is directly assigned as many copies as indicated by the integer part of Ei in the new population. If the intermediate population has not yet been completed, the original population is sorted in descending order of the fractional parts of Ei, and we choose solution by solution, in that order, until the new population is completed.

In either version of *stochastic sampling,* which is with or without replacement, each solution is directly assigned as many copies as indicated by the integer part of its expected value, as in the previously mentioned case. However, they differ in how they treat the fractional parts of these expected values. In the version with replacement, fractional parts are used to form a wheel, as in simple selection, and the ball will be released as many times as solutions are needed to complete the new population. In the version without replacement, the fractional part is used as the probability of getting another copy. That is, in this type of selection a solution with the expected value of 1.8, would receive a copy safely and will have a probability of 80% of obtaining a second copy.

Previous methods of selection, in which the number of copies that a particular individual gets depends on its relative value of adequacy with respect to the rest of the population, may present the following problem. During the execution of the genetic algorithm, in the first iterations there are usually some super individuals, with a much better fitness value than the rest. By applying the above selection processes as such, these individuals would represent a large proportion of the population in very few iterations. This is undesirable because of the loss of diversity in the population and it would possibly result in a premature convergence. Furthermore, in the last iterations, the opposite effect occurs. Once the population has evolved considerably and improved, the average fitness value of the population is possibly close to the value of the best and worst individuals. That is, since there are no major differences among individuals, the best individuals would receive about the same number of copies as the worst ones. In this way the search, which based on the principle of survival of the fittest individuals needed to properly steer population improvement in one direction, becomes a blind search or random search.

To avoid the above problem *scaling* techniques may be used additionally. These techniques modify the fitness values of individuals in the population (fitness values are scaled) so that the number of copies that the best solutions obtain is limited in the first and the latest iterations. The best individuals receive, in ratio, a higher number of copies than the less qualified, although their fitness values do not differ significantly.

A different type of selection mechanism which does not present the above problem is given by mechanisms based on the ranking of individuals in the population. The individuals in the population are sorted in increasing order of their fitness values. The number of copies is assigned to each individual in proportion to their place in the ranking, rather than being proportional to its relative fitness value.

One selection mechanism that combines the ideas of the methods discussed above together with the idea of ranking based methods is the so called *tournament selection*. This method selects k individuals in each iteration using the roulette method. Of these k solutions, only the one with the best fitness value is copied into the new population. This process is repeated *pop_size* times, i.e. until the new population is complete. Obviously higher k values increase the pressure of the selection process. An accepted value which has given good results in various problems is k=2. That is, in each iteration we select two individuals and only the best of them becomes part of the new population.

9.1.4. CROSSOVER

The crossover process is applied to the population resulting from the selection process. The solutions are randomly paired and each of these pairs undergoes the crossover operation with a certain crossover probability Pc, which is fixed for the entire population. If the couple does not participate, both of the individuals in the population remain unchanged. By contrast, if the pair performs the crossover, the mechanism creates two new solutions which inherit a combination of features of their parents and replace them. Thus, after the crossover process, the size of the population remains unchanged.

Due to the selection process, although such pairs are chosen randomly, the best individuals are involved in the crossover process more actively, because they appear in the population more often, and they are therefore the ones with the higher probability of passing on their genetic material to successive generations. Also, due to the fact that if one set of parents does not carry out the crossover, they remain unaltered, individuals from different generations can live together in the same population, as occurs in nature.

Crossover technique should be designed carefully so that the new solutions, the offspring, may be created by combining the characteristics of the two solutions involved in the process, that is, the parents. Crossover process should not consist solely of a combination of the solutions to form new solutions. In addition this combination should be truly beneficial. There are several ways to carry out the crossover process some of which we will outline.

The simplest mechanism which has been applied in a variety of problems providing excellent results is the *one-point crossover*. In this technique, and given two solutions of length l, we randomly choose a crossover point k, with $1 \le k < l$, such that in one of the resulting solutions, “the daughter”, the first k genes are inherited from one parent, for example the mother, and the remaining genes from the other, in this case the father. The other solution, “the son”, inherits the first k genes from the father and the rest from the mother.

Figure 9.2 shows how this type of crossover works over a simple example. Let us suppose solutions represented by strings of length 10, defined over the binary alphabet {0, 1}. The crossover point, selected randomly could be *k=4*. Thus, the daughter solution inherits the first four genes from the mother solution and, the rest of the genes, those placed between positions 5 and 10, from the father. In the same way, the son inherits the first four genes from the father, and the remaining 6 from the mother. So the two children inherit characteristics from both parents.

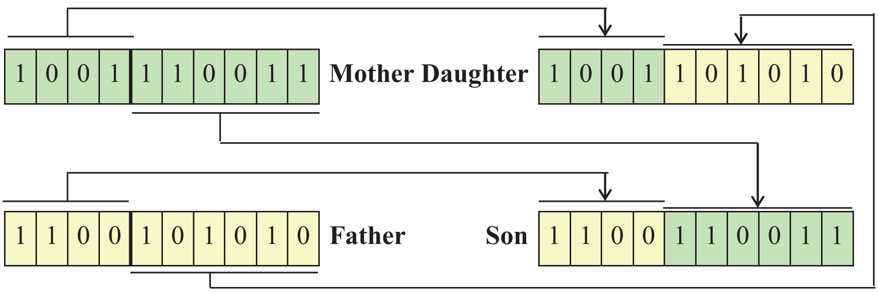
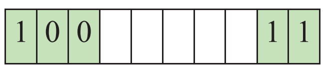


Figure 9. 2. One-point crossover example

One problem with this procedure is the following: let us suppose that the following combination of genes that the mother presents, is a combination that contributes decisively to the good performance presented by this solution:



Now, with the one-point crossover mechanism, whatever the generated crossover point may be, it would be impossible for any child to inherit this gene combination present in the mother. That is, there are sets of features that, even though present in the parents cannot be inherited by the offspring.

Therefore, a technique in which the number of crossover points is two instead of one has been developed in order to try to remedy this problem. This technique is called 2-point crossover. In this procedure, the crossover points, *k1* and *k2*, are also randomly generated so that $1 \leq k1 < k2 \leq l$. Now the daughter solution will inherit the mother´s first *k1* genes, the ones located between *k+1* and *k2* from the father, and the genes between *k2 +1* and *l* again from the mother. By contrast, the son inherits the father's first *k1* genes, the genes located between *k1+1* and *k2* from the mother, and the genes located between positions *k2+1* and *l* from the father.

Let us consider the functioning of this technique applied to the same example that we have previously used with the one-point crossover. This example is shown in Figure 9.3. Suppose now that the two crossover points generated at random, were *k1=4* and *k2=7*.

This example shows that the characteristics of the mother that could not be inherited by any offspring using the technique of one-point crossover, have been acquired by the daughter with this other type of crossover. That is, with the 2-point crossover a solution can inherit combinations of genes present in the parents that are not inheritable with the one-point crossover.

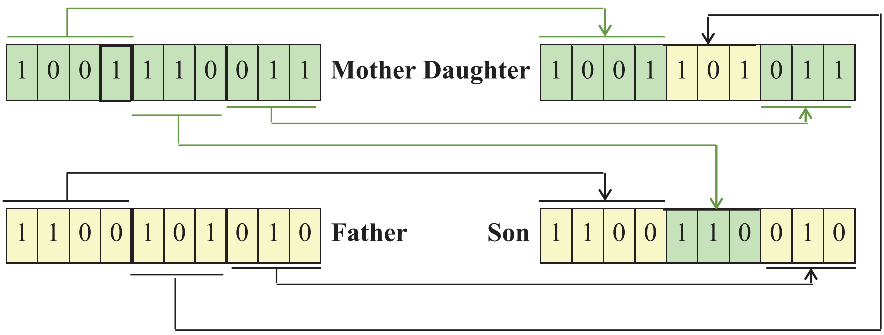


Figure 9.3. Two-point crossover example

In the same way, if you apply the 2-point crossover, there could be combinations of genes that are present in the parents and are not inheritable by the offspring. For this reason, the one-point crossover can be generalized so that there are *t* crossover points, yielding the so-called *multi-point crossover*. Special cases of multi-point crossover would be then the one-point crossover (*t=1*) and the 2-point crossover (*t=2*).

A different crossover mechanism is the *uniform crossover*. To apply it, the technique first generates what is called cross mask, which consists of a string of the same length as the solutions, with ones and zeros randomly generated. A random mask is created for each pair to cross. To generate one offspring, such as the daughter, the way to proceed is: If at position *i* of the mask there is a 1, the daughter inherits the gene from the mother, and if there is a 0, the gene is inherited from the father. To generate the son the procedure is analogous but the parents are exchanged. An example is shown in Figure 9.4.

Several authors have compared the behaviour of the 2-point crossover with the uniform crossover and none of them seems to always have better behaviour for all problems.

1	0	0	1	1	1	0	0	1	1	**Mother**
0	1	0	1	0	0	0	1	1	0	**Crossover mask**
1	1	0	0	1	0	1	0	1	0	**Father**

Daughter	1	0	0	1	1	0	1	0	1	0
Son	1	1	0	0	1	1	0	0	1	1

Figure 9.4. Uniform crossover example

There are other crossover techniques that are radically different from the previously addressed ones. One of these is the *PM*X crossover (Partially Matched Crossover) proposed by Goldberg and Lingle (1985), indicated for those problems in which the fitness value of a solution depends only on the order in which the genes appear, and their values are fixed. In this type of problem, the solutions are represented by permutations of the elements of the alphabet. An example of this type of problem could be the Traveling Salesman Problem (TSP). A traveller must visit *n* cities so that the distance or the total costs are minimized. In this case, the solutions may be represented by chromosomes with as many genes as cities, so that if, for example, the value "Madrid" appears at position *i*, it means that Madrid would be the i-th city to be visited. If, in this problem, with the coding of the solutions indicated above, we apply any of the techniques previously described, the result would probably be a non-feasible solution.

Consider the application of *PM*X crossover to a specific example, shown in Figure 9.5. As in the 2-point crossover, the first step in *PM*X is to randomly generate two points $k1$ and $k2$, so that $1 \leq k1 < k2 \leq l$. Each child inherits the genes contained between positions $k1$ and $k2$ from one parent. Both positions define both a set of exchanges which will be used later. Let us suppose in our example that $k1$=4 and $k2$=7:

1	3	4	6	2	5	8	7	9	**Mother**
?	?	?	?	2	5	8	?	?	**Daughter**

↕ ↕ ↕

3	2	5	7	1	9	4	6	8	**Father**	**Son**	?	?	?	?	1	9	4	?	?

Figure 9.5(a). *PM*X crossover example: 1st step

The defined interchanges have been marked with arrows in the previous figure. These are:

2 ↔ 1   5 ↔ 9   8 ↔ 4

The next step is to inherit those genes marked by ?, from the other parent and do not produce conflict with each other, i.e. they do not appear repeatedly. It means that the daughter now inherits genes from the father and the son from the mother.

1	3	4	6	2	5	8	7	9	**Mother**	**Daughter**	**3**	?	?	**7**	2	5	8	**6**	?
3	2	5	7	1	9	4	6	8	**Father**	**Son**	?	**3**	?	**6**	1	9	4	**7**	?

Figure 9.5 (b). PMX crossover example: 2nd step

Finally, the exchanges defined above serve to inherit the rest of the genes, which are those that could not be inherited in the previous step by producing conflicts:

1	3	4	6	2	5	8	7	9	**Mother**	**Daughter**	3	**1**	**9**	7	2	5	8	6	**4**
3	2	5	7	1	9	4	6	8	**Father**	**Son**	**2**	3	**8**	6	1	9	4	7	**5**

Figure 9.5 (c). PMX crossover example: 3rd step

In the previous step, the daughter should have inherited a "2" in the second position from the father, but this would have caused a conflict since the "2" appeared in the daughter in the 5th position. Therefore, we used the previously defined set of exchanges whereby the "2" has been exchanged for a "1", which is finally displayed on the second position of the daughter. The same applies to the third and final positions of the daughter. This fact is also given in the first, third and last positions of the son.

Another crossover which has been defined for this type of problems is the *order crossover* proposed by Davis (1985). As in the previous crossover, the first step consists of randomly generating two crossover points *k1* and *k2* so that $1 \leq k1 \leq k2 \leq l$. In the same way as in the *PM*X, each of the descendants inherits the genes contained between positions *k1* and *k2* from one parent. The difference from the previous crossover is the following: the remaining genes are now inherited by the offspring in the relative order in which they appear in the other parent. Let us see the application in the next example. Suppose again that *k1*=4 and *k2*=7:

1	3	4	6	2	5	8	7	9	**Mother**	**Daughter**	?	?	?	?	2	5	8	?	?
3	2	5	7	1	9	4	6	8	**Father**	**Son**	?	?	?	?	1	9	4	?	?

Figure 9.6(a). Order crossover example: 1st step

Now the daughter will inherit the genes marked as ?, in the relative order in which they appear in the father. That is, the daughter has only inherited the genes "2", "5" and "8", so she must inherit the rest. These will be inherited one by one keeping the relative order in which they appear in the father. The relative order of these genes in the father is 3 - 7 - 1 - 9 - 4 - 6:

1	3	4	6	2	5	8	7	9	**Mother**	**Daughter**	3	7	1	9	2	5	8	4	6
3	2	5	7	1	9	4	6	8	**Father**	**Son**	?	?	?	?	1	9	4	?	?

Figure 9.6 (a). Order crossover example: 2nd step

The main difference between the first three crossover techniques exposed (one-point crossover, two-point crossover and multi-point crossover) with respect to the mentioned later (*PM*X and order crossover) is that the latter ones incorporate problem-specific knowledge, while in the former, no reference has been made at all to the type of problem they solve. One of the advantages of incorporating problem-specific knowledge is that non feasible solutions which are sometimes difficult to manage can be avoided (as could have happened in the traveling salesman problem if we had used any of the first three techniques).

9.1.5. MUTATION

Once the crossover process has finished the mutation procedure is applied. The purpose of this process is to **introduce some variability** into the population, introducing some new features or characteristics in individuals, which were in the population in the past, but were lost during the process of evolution. That is, the mutation, unlike the process of crossover, is a kind of blind search that seeks to ensure that in the solution space there is no point with a zero probability of being examined.

Mutation affects each of the genes of each of the individuals in the population with a certain probability, the mutation probability, P_m, which is usually quite small (much smaller than the crossover probability). That is, each and every one of the genes has equal probability of being affected by the mutation. Mutating a gene consists of altering its value.

The simple mutation consists of exploring each individual, and for each of its genes, generate a random number between 0 and 1, so that if the number is less than or equal to P_m, then the value of that gene is chosen randomly among the remaining symbols of the alphabet. As with the crossover operator, mutation may also incorporate problem-specific knowledge.

9.1.6. APPLICATIONS: THE TRAVELLING SALESMAN PROBLEM

Since the metaheuristic techniques in general and the genetic algorithms in particular are suitable to be used in combinatorial optimization problems, they have many applications in the field of decision making in administration and business management. Below there are some examples of genetic algorithms that have been successfully designed and applied to solve business and management problems such us: stock exchange index prediction, collection routes for freight vehicles, resource allocation in project scheduling, allocation of human resources for construction, shifts, inventory management, economic predictions, etc.

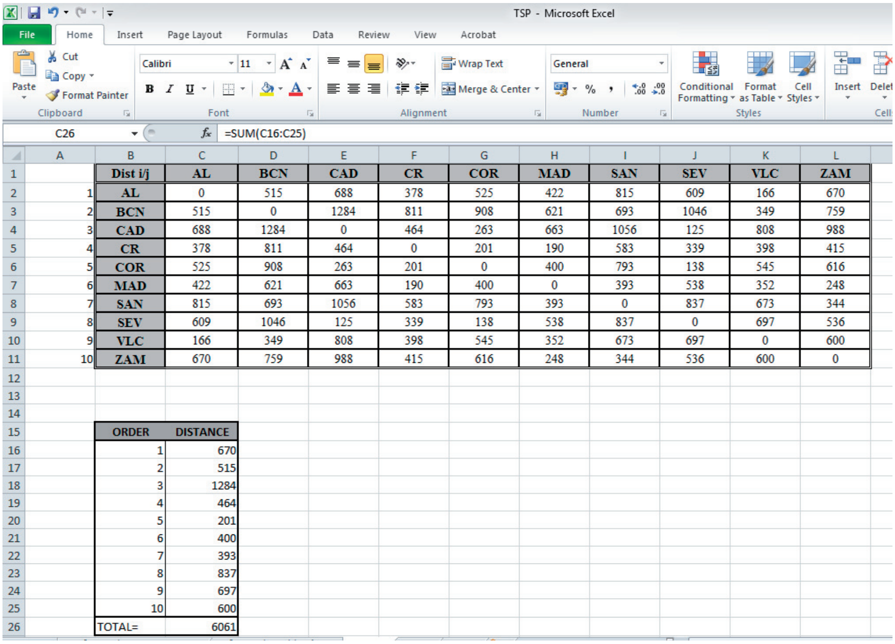
The travelling salesman problem is one of the classic problems in Operational Research in which a salesman must visit a number of cities and he wants to find the optimal path that minimizes the total distance to travel. Take the following simple example.

A travelling salesman of the ADESA company has to plan visits to be made to some of his customers over the following week. There are ten customers to visit and they are located in Alicante, Valencia, Sevilla, Cádiz, La Coruña, Santander, Madrid, Barcelona, Ciudad Real and Zamora. Figure 9.7 shows the distance between each pair of cities on the usual route used by the traveling salesman (road, highway, freeway, etc.).

We then seek a cyclic route for him to visit every city once and finish in the same city where he started. Thus, the same route would be valid for any of the starting cities. For example, if the optimal route was

AL→ VLC→BCN→CR→SAN→COR→SEV→CAD→ZAM→MAD

we could choose any of the cities as the starting city. If we start from Alicante, the first city to visit will be Valencia and then Barcelona, Ciudad Real, Santander, Córdoba, Sevilla, Cádiz, Zamora, Madrid and back to Alicante. If we start from Córdoba instead, we will go in the first place to Sevilla and then to Cádiz, Zamora, Madrid, Alicante, Valencia, Barcelona, Ciudad Real, Santander and back to Córdoba. Obviously, the total distance travelled in both routes is exactly the same.



	Dist i/j	AL	BCN	CAD	CR	COR	MAD	SAN	SEV	VLC	ZAM
1	AL	0	515	688	378	525	422	815	609	166	670
2	BCN	515	0	1284	811	908	621	693	1046	349	759
3	CAD	688	1284	0	464	263	663	1056	125	808	988
4	CR	378	811	464	0	201	190	583	339	398	415
5	COR	525	908	263	201	0	400	793	138	545	616
6	MAD	422	621	663	190	400	0	393	538	352	248
7	SAN	815	693	1056	583	793	393	0	837	673	344
8	SEV	609	1046	125	339	138	538	837	0	697	536
9	VLC	166	349	808	398	545	352	673	697	0	600
10	ZAM	670	759	988	415	616	248	344	536	600	0

ORDER	DISTANCE
1	670
2	515
3	1284
4	464
5	201
6	400
7	393
8	837
9	697
10	600
TOTAL=	6061

Figure 9.7. Distances between cities

The above problem has many other applications such as the collection of money in city phone boxes, the establishment of a fibre optic line between a group of users or the delivery of pizzas of a particular dealer.

There are several alternatives to solve the above problem. The first is to present a binary linear programming model. There are different possibilities for its formulation. In one of them, the resulting model would have a total of 100 binary variables and 101 constraints. In general terms, with this formulation, the mathematical model would have a total of n^2 variables and $n^2 +1$ constraints, where n is the number of cities. In many cases, these models are not solvable in practice, making it necessary to employ heuristics or metaheuristics. Let us see how to design a genetic algorithm to solve the above mentioned problem.

Solutions encoding

To design an appropriate encoding for the solutions to the problem, we have to think about how to express a solution to it. In this problem, a solution can be represented by the order in which the cities should be visited, for example:

MAD→VAL→AL→CAD→SEV→COR→CR→ZAM→SAN→BCN

would be a way of expressing a solution, considering that from Barcelona we would finally return to Madrid.

Thus, any permutation of the cities, represents a feasible solution to the problem. One way to represent the solutions would be to encode them using ordered lists in which each gene is one of the cities to visit. The above solution can be expressed as:

MAD	**VLC**	**AL**	**CAD**	**SEV**	**COR**	**CR**	**ZAM**	**SAN**	**BCN**

To simplify the notation, from now on we will represent each city by their order in the list of names in alphabetical order, so that the above solution would be represented as:

6	**9**	**1**	**3**	**8**	**5**	**4**	**10**	**7**	**2**

With the above encoding, any list that consists of a permutation of cities represents a feasible solution to the problem. We want to obtain the best, ie the optimal solution among all the feasible solutions. There are n! different permutations for a problem with n cities. In this case we would have a total of 10!=3,628,800 possible orderings.

Initial Population

Since any permutation of the cities represents a solution of the problem, a simple and low computational effort method to create the initial population would consist of generating random solutions. An alternative could be to apply heuristic algorithms that gave us solutions to the problem and use these as initial population. Both methods could also be combined to generate the initial population.

An important parameter to consider in the algorithm is the size of the population that we are going to manage. It is common to use a standard size of 50 to 100 individuals, although other sizes may be advisable. The size is often determined after some preliminary tests.

Evaluation

It is necessary to define a function that assigns a fitness value for each solution, which indicates the value of the solution in relation to the objective. In our case, the objective is to minimize the total distance travelled along the route, so that the best solution would be the one whose total distance is minimal.

6	**9**	**1**	**3**	**8**	**5**	**4**	**10**	**7**	**2**

Solution represents a route in which the total distance is 3743 kilometres. The total distance travelled could be the direct fitness value, so that the best individuals would be those who have a lower fitness value. However, many standard mechanisms (mainly selection ones) used in genetic algorithms are designed to maximize the fitness values. Therefore, we could transform the fitness values of the individuals so that the best individuals have the highest fitness values and not the lowest. That is, the optimal solution of this problem should have the highest possible fitness value. One way of doing it, amongs many others, would be to establish an upper bound on the total distance travelled and to subtract the value of this distance bound for each individual in the population. One way to establish an upper bound on the distance travelled would be to choose the greatest distance from each of the columns of the table and add them. Thus, this upper bound would be given by:

OF_Upper_bound = 815+1284+1284+811+908+663 +1056 +1046 +808 +988=9663

Now, the fitness value of each individual would be obtained by subtracting, from the previous bound, the distance of the route that it represents. Thus, we would get an adequacy value of 9663-3743=5920 for the solution shown above, which represented a route of 3743 kilometres. The solution

2	**4**	**6**	**8**	**10**	**1**	**3**	**5**	**7**	**9**

representing a route of 5511 kilometres, would have a fitness value of 4151 lower than the previous one. The optimal solution would have the highest fitness value and the worst solution would have the lowest fitness value among all possible.

Selection

Any selection mechanism outlined in the previous section could be used in this example.

Crossover

For this problem and with the solutions encoded in this way, we could apply any type of previously defined crossover mechanism that works for permutations, for example *PMX* crossover or *order crossover* both presented in section 9.1.4 of this chapter. We could also define a specific type of crossover for this problem, taking into account their special conditions and incorporating them to generate the offspring in an efficient way.

We should also set the crossover probability. Values around 80-90% are often used, but, as with the population size, it is usually set after performing some tests.

Mutation

The mutation involves making random changes to a solution mimicking the mutations that occur on the genetic material of living creatures in nature. Such mutations could make desirable or undesirable attributes appear in individuals. It moves through the solution and each gene is mutated with a certain probability of mutation, which is usually low, at around 1%, although it can vary depending on the problem. For this problem, mutating a single gene could consist of inserting it at any position of the chromosome, or exchanging it directly for the front or rear ones.

A simple genetic algorithm such as the one described in the preceding paragraphs, is able to solve sizable instances of this problem in very reasonable computation times obtaining very good solutions which makes these techniques the best alternative to solving some real problems.

To solve this problem with Solver, we have two options. The first is to build a linear programming model and use the method "Simplex LP" to solve it. It should be borne in mind that the resulting linear model can be large and the time required for solving it can be extremely long. Due to its size it may be unsolvable even on high-end workstations. Furthermore, we have to take into account the limitation in the number of variables and number of constraints of the Solver version employed. In particular, the default version in Excel solves linear models with up to 200 variables, with no limit on the number of constraints. We could not solve problems with 15 cities or more by formulating the binary programming model discussed above with Solver.

The second option would consist of employing the "Evolutionary" method, that uses genetic algorithms and is especially designed for unsmoothed type problems. Let us see how to easily define the problem and then find out now to use the "Evolutionary" method supported by Excel Solver to solve it.

First we define the distance matrix, as shown in Figure 9.7. This matrix is symmetric and zeros always appear in its diagonal. Then we should prepare the cells in which the solution to the problem will be displayed once solved and how to calculate, based on the solution, the value of the objective function, ie, the total distance in kilometres.

We will use two columns, ORDER and DISTANCE. The first column is left blank, because cells B16: B25 are used to represent the solution once the problem has been solved. That is, the value in cell B17 indicates the number of the city to visit after having visited the one whose number appears in cell B16 and so on. The second column will serve to calculate the total distance travelled by the salesman if you follow the order given in the solution indicated by the column ORDER. To obtain this order, we will use the INDEX function.

The syntax of this function is:

= INDEX (array, row_num, column_num)

and it returns the value of an item in a table or matrix selected by the row and column indices given. Thus, in cell C17 we want to calculate the distance from the city whose number appears in cell B16 to the city whose number appears in cell B17, for which we write =INDEX ($C$2:$L$11; B16; B17) in that cell and drag the formula to cells C18 to C25.

In cell B16 we write =INDEX (C2:L11; B25; B16). If Column B has been left blank, # VALUE! will appear in these cells when writing the formulas, since we are using numerical values that do not exist in these formulas. This is not a mistake. If we fill in cells B16: B25 with a list of values between 1 and 10 (a feasible solution to the problem) we see that in column C the corresponding distance between each pair of cities on the route is displayed.

Finally, in cell C26 we calculate the sum of the distances travelled by typing =SUM (C16: C25). The value displayed in the cell is the total distance travelled, which we want to minimize.

We can now press the button to open the Solver dialog box and define the model to be solved, based on the sheet set, as shown in Figure 9.8.

Set Objective: in this case, the total distance travelled is displayed in cell C26.

To: Our goal is to minimize the total distance.

By Changing Variable Cells: On the sheet we have reserved space for the variables in cells B16 to B25

Subject to the Constraints: Clicking the Add button, Solver displays the dialog box to add constraints:

Cell Reference: select the range of cells that represent the variables, ie the range B16: B25

In the centre of this dialog box, which is normally used to indicate the sign of inequality or the character of the variables, we select *diff*, which indicates that in these cells Solver must calculate a permutation of the integer values that are consecutive and start at 1, that is, a permutation of the integers between 1 and 10 in our case.

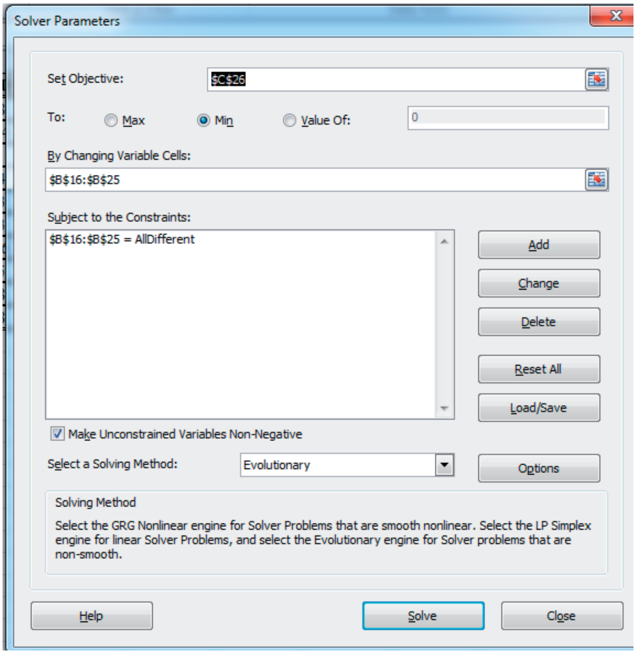


Figure 9.8. Solver Parameters

Select a solving method: we select "Evolutionary" to apply the mechanism based on genetic algorithms. Moreover, the way in which we have defined the problem makes it impossible to use the LP Simplex method. Then, before solving the problem, we can configure this method, for which we must press the *Options* button and go to the tab "Evolutionary", which displays the dialog box shown in Figure 9.9.

Next we set the values shown in Figure 9.9 for the different parameters (as an exercise, the student could try to change the values of the different parameters and observe the effect they have on the solution and the time it takes to find it).

Once we click *Solve*, Excel starts the search for the best solution. During the search process, in the bottom bar of the screen, the method displays information about the process, such as the total number of solutions evaluated and the objective value of the best solution found so far. We can interrupt the search process by pressing the *Esc* key at any time. In this case, Solver displays the best solution found so far. The best solution found by "Evolutionary" Solver for this example is shown in Figure 9.10, representing a total distance of 3142 kilometres. So, if the traveller started, for example, from Valencia

(City 9) the route to follow would consist of visiting Barcelona first, and then going to Santander, Zamora, Madrid, Ciudad Real, Córdoba, Sevilla, Cádiz, Alicante and returning to Valencia. The reverse route would be equivalent, with the same distance travelled. As the solution is given by a metaheuristic technique, we cannot generally know whether it is the optimal solution to the problem or not. In this case we can ensure that it is, because we have optimality solved the problem by other methods and the optimal solution is already known.

If Solver finishes its execution without being interrupted, it permits the user to generate two different reports, called *Answer* and *Population*, which are presented in Figures 9.11 and 9.12.

In the *Answer* report, the software shows the options used by the resolution method, the total time and the number of solutions evaluated. It also indicates the value of the target cell and the best value of the variables.

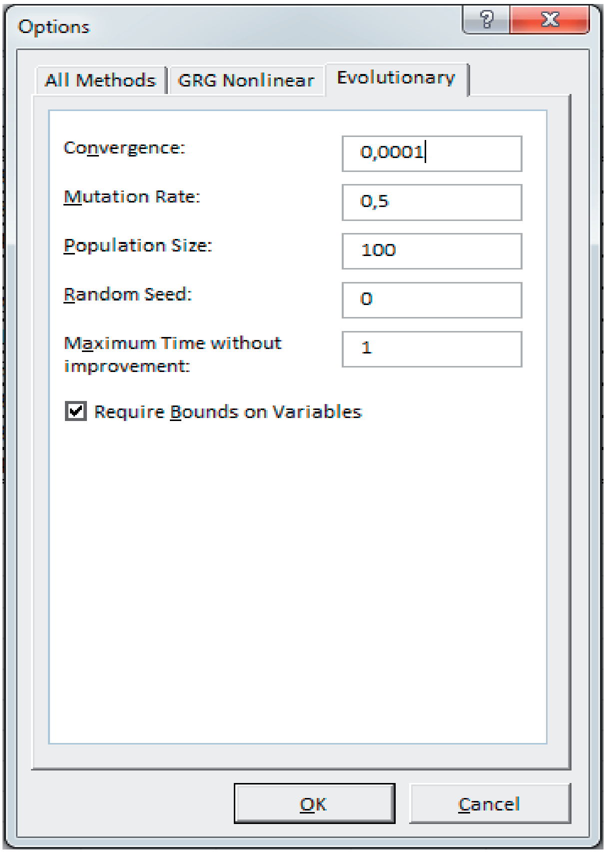


Figure 9.9. Options of Evolutionary Solver

ORDEN	DISTANCIA
3	125
1	688
9	166
2	349
7	693
10	344
6	248
4	190
5	201
8	138
TOTAL=	3142

Figure 9.10. Best solution

The *Population* report shows data about the solutions which the method finds during the search process until it finds the best solution. In this case it reports the minimum and maximum values that are assigned to each of the variables, as well as their mean value and standard deviation.

Result: Solver cannot improve the current solution. All Constraints are satisfied.

Solver Engine

Engine: Evolutionary

Solution Time: 8,705 Seconds.

Iterations: 0 Subproblems: 3569

Solver Options

Max Time Unlimited, Iterations Unlimited, Precision 0,000001

Convergence 0,0001, Population Size 100, Random Seed 0, Mutation Rate 0,5, Time w/o

Max Subproblems 10000, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume No

Objective Cell (Min)

Cell	Name	Original Value	Final Value
$C$26	TOTAL= DISTANCE	6061	3142

Variable Cells

Cell	Name	Original Value	Final Value	Integer
$B$16	ORDER	1	6	AllDiff
$B$17	ORDER	2	4	AllDiff
$B$18	ORDER	3	5	AllDiff
$B$19	ORDER	4	8	AllDiff
$B$20	ORDER	5	3	AllDiff
$B$21	ORDER	6	1	AllDiff
$B$22	ORDER	7	9	AllDiff
$B$23	ORDER	8	2	AllDiff
$B$24	ORDER	9	7	AllDiff
$B$25	ORDER	10	10	AllDiff

Constraints

NONE
$B$16:$B$25=AllDiff

Figure 9.11. Evolutionary Solver: Answer report

Variable Cells

Cell	Name	Best Value	Mean Value	Standard Deviation	Maximum Value	Minimum Value
$B$16	ORDER	6	5,55	2,868762447	10	1
$B$17	ORDER	4	4,78	3,066864289	10	1
$B$18	ORDER	5	6,44	2,660295419	10	1
$B$19	ORDER	8	5,41	3,547925842	10	1
$B$20	ORDER	3	5,5	2,231546096	10	1
$B$21	ORDER	1	5,52	2,572111501	10	1
$B$22	ORDER	9	5,7	1,982244417	10	1
$B$23	ORDER	2	5,45	2,875795893	10	1
$B$24	ORDER	7	5,07	2,843582177	10	1
$B$25	ORDER	10	5,58	3,533905042	10	1

Figure 9.12. Evolutionary Solver: Population report

9.2. TABU SEARCH

The roots of this technique date back to the 70s, but was first introduced by Fred Glover (1986) in its current form and later refined in "Glover and Laguna (1997)". Although there is no theoretical evidence of its convergence, many computational experiments have shown that this is a technique that can compete with the best existing metaheuristics, and due to its flexibility, it has been successfully applied to a wide variety of problems.

Tabu search is a metaheuristic procedure that guides a local search procedure to explore the solution space beyond local optima. It uses procedures that are designed to push the limits of feasibility (to try to avoid being trapped in local optima), which were normally treated as insurmountable barriers.

A good analogy to this technique could represent the procedure followed by a trained climber to reach the top of a mountain. The climber must choose an appropriate path to follow at all times. Each of these choices should be based on knowledge gained during the ascent. The climber must analyse the different alternatives in relation to the steps followed earlier in similar parts of the mountain at all times. Perhaps at some point it could be advisable to descend a certain amount if this setback is somehow compensated for later on. Also, if the climber does not make use of his memory and remember the points that he has gone through, he could be stuck in a cycle: from point "a" he goes to "b" from "b" to "c" from " c "to" a "...

It must be said that not only local information (such as the value of the objective function at the point where we are), but also information on the scanning process should be used to improve the efficiency of the exploration of the solution space. The **systematic use of memory** is an essential aspect of this technique. Although many methods of exploration "remember" the objective value of the best solution found so far, tabu search (TS) also holds in memory the **path followed by the last visited solutions**.

Tabu search starts in the same way as conventional methods of local search do, proceeding iteratively and sequentially from one point to another until a termination criterion is satisfied. Each solution has an associated neighbour set of solutions *V(s)*, and a solution $s' \in V(s)$ is reached from *s* by an operation m that we call movement. Thus, set *V(s)* is obtained by applying all possible moves to *s*. The m's complementary movement would be that movement m' that would allow us to go from *s'* to s.

The general idea is to build the set *V(s)* for the current solution s and choose from this set the best individual as the new current solution in every iteration. *V(s)* can be large and therefore expensive to evaluate, so we usually reduce the set to a subset of it, $V^*(s)$. Since it is possible that the new solution is even worse than the previous one, there is a risk of visiting a solution more than once and, more generally cycles can take place. Memory is then used to try to avoid this, prohibiting a movement that would lead to recently visited solutions. Thus, we can consider that set *V(s)* depends not only on s, but also on the time *t* of the search process in which we are, so from now on we will refer to it as *V(s,t)*. That is, some recently visited solutions that belong to *V(s)* are removed to form *V(s,t)*. These solutions are considered tabu solutions.

Using a list L of length l, whose elements are the latest solutions visited in each iteration, we could build *V(s,t)* by eliminating from *V(s)* the solutions found in L. This will avoid cycles of length l. As this list may not be manageable in practice, we can substitute it with a list L of the movements that are complementary to the recent ones performed. Therefore, and using these tabu movements, solutions that are accessed from *s* will not appear in *V(s,t)*.

The simplification from the list of solutions to the list of movements means a loss of information and therefore it has a drawback. Prohibited movements could make it impossible to visit solutions that have not been visited so far. For this reason, the process allows the application of a tabu movement to obtain a given solution if a certain aspiration level is reached. Aspiration levels define desirable conditions in the solutions, so that if a particular solution meets any of these levels, it can be transformed from tabu solution to allowed solution. Moreover, such aspiration levels may be variable over time.

Another important aspect of this technique is the relationship between intensification and diversification. At certain times of the search process it may be advisable to focus the search on the current region of the solution space, because it contains acceptable solutions. This enhancement can be achieved by prioritizing the solutions that have features in common with the current solution and penalizing those that are farther away. At another moment of the search process, maybe we want to move away from the current region and focus on another area, perhaps because the current region contains no more acceptable solutions or because the other region seems more promising. This diversification can be achieved by penalizing solutions which are similar to the current solution and giving priority to those that differ more from it. We can therefore replace the original problem objective function f, with a new objective function $\overline{f}$ that incorporates these features. The weight assigned to intensification and diversification will be variable and will be modified depending on the needs that occur during the search process.

The algorithmic scheme of this technique is presented in pseudocode in Figure 9.13.

```
Tabu Search Procedure
  Generate_initial_solution (s)
  Best =s
  t=0
  while not (termination criterion)
    Generate V*(s,t) ⊆ V(s,t)
    s*=best of V*(s,t)
    st=s*
    if s* is better than Best
      Best=s*
    t=t+1
```

Figure 9.13. Tabu Search: General procedure

The procedure Generate $V^{*}(s\_act, t) \subseteq V(s\_act, t)$ will take into account the list of tabu movements and aspiration levels described above. In addition to choosing the best element of a set, and establishing whether a solution is better than another, we also take into account function $\overline{f}$, which incorporates intensification and diversification mechanisms.

After applying this algorithm, the best solution found is offered as the solution to the problem. We can define, among others, the following stopping conditions:

- The set $V(s\_act, t)$ is empty.
- A number of iterations have been completed.
- In the last k iterations the best solution found has not changed.
- An acceptable solution has been found.

Thus, tabu search is defined as a sequential search metaheuristic algorithm, since only one solution is chosen in every iteration, and which incorporates a memory element for guiding the search process.

9.3. SIMULATED ANNEALING

Simulated annealing is considered as a metaheuristic algorithm that can be successfully applied to a wide variety of problems. It also has a stochastic component that provides the theoretical analysis of its convergence, which has made it become a very attractive method.

The ideas that form the basis of this technique were published by Metropolis et al (1953) as part of an algorithm, which simulates the process that a solid material undergoes when introduced into a liquid at high temperatures until it melts and then it is allowed to cool slowly to solid state. This process is known in physics as "annealing". When the solid is heated to melt, and then cooled again to its transformation into a solid, the structural features depend on the cooling rate, as during the heating phase the particles are arranged at random, and therefore, depending on the time it takes to cool, they can be arranged in one way or another. The process of "annealing" can be simulated by considering the solid material as an assembly of particles. Metropolis successfully simulated this process on a computer using Monte Carlo techniques for this purpose.

Thirty years later Kirkpatrick et al. (1983) proposed that this type of simulation could be used to solve optimization problems. In this technique, which is a sequential technique like tabu search, only one solution is obtained from a previous solution at each iteration and the process is repeated until a termination condition is satisfied.

If we denote, as in tabu search, $V(s)$ as the set of solutions that can be obtained from the solution s, the process is as follows: starting from an initial solution s, we choose, usually at random, any solution $s' \in V(s)$. This solution s' is accepted as a new current solution if it is better than s. However, if s' is worse than s, there is also a chance of it being accepted as the current solution. This probability depends on both the objective value given by the solution and the parameter t, known as *temperature* by analogy with the physical process described above. In the first moments of the process, the temperature is high, and so is the probability of accepting any solution. As the process progresses, the temperature decreases, and this probability also decreases.

That is, in the initial moments of the process, when the temperature is high, the search is a kind of random search, since virtually any solution, good or bad, is accepted. In such moments the search is dominated by the exploration of the solution space. However, as the process progresses and the temperature drops, the exploration of the solution space is being replaced by exploitation, since now the search is guided almost exclusively by the best solutions.

The choice of the parameter t has a significant influence on the behaviour of the algorithm. Its initial value and the manner in which this value decreases along the process must be set, this is called *cooling program*. If cooling is too fast, the technique tends to behave as a simple local search mechanism and can be trapped in a local optimum or a low quality solution. Moreover, if the cooling is too slow, the running time of the algorithm can become prohibitive.

Apart from the temperature parameter definition, the probability of acceptance of a new solution $s' \in V(s)$ must also be set, where s is the current solution. As discussed above, this probability depends on the objective value of the new solution (or its difference from the value of the current solution) and the current temperature, t. If we denote by Δ the difference between the objective values of the current solution s and the new solution s', defined as $\Delta = f(s') - f(s)$, and assuming that we minimize the objective function value, the probability of accepting s' might be defined by:

$$P(s \rightarrow s') = \begin{cases} 1 & if \quad \Delta < 0 \\ e^{-\Delta / t} & if \quad \Delta \geq 0 \end{cases}$$

The above formula would be a possible definition of the probability of acceptance. There are many other alternatives and the researcher should be the one responsible for choosing it properly, as was the case of the temperature parameter. Figure 9.14 shows the basic operation of the technique in pseudocode.

The first step consists of generating the starting solution either randomly or by some heuristic procedure. The temperature parameter, t, must also be initialized with a certain value. Then, these steps are repeated until the termination criterion is satisfied. The set of solutions accessible from the current one is generated, and a solution is chosen from this set. In its simplest form, such a solution is chosen randomly, although another mechanism could be used. If this solution is better than the one we had, it is accepted as current solution, if not it is accepted with a certain probability of acceptance. Finally the temperature is decreased.

Once the termination condition has been satisfied, the current solution is proposed by this technique as the solution to the problem. Some of the completion criterion that can be set are:

- The set $V(s)$ is empty.
- A number of iterations have been completed.
- The temperature reaches the minimum value (usually 0)
- An acceptable solution has been found.

Finally, the most remarkable difference between simulated annealing and tabu search is the use of the memory by the latter. Recently, some variations of the technique, known as hybrid techniques, have appeared. They incorporate the use of the memory to improve the efficiency of the technique in an effective way, among other new features.

```
Simulated Annealing Procedure
   Generate_initial_solution(s)
   t=init_temp
   while not (termination_criterion)
       Generate V(s)
       s' =choose solution from V(s)
       if s'  is better than s
            s=s'
       else
            P=calculate P(s-->s',t)
       s=s'  with probability P
       update t
```

Figure 9.14. Simulated Annealing: general procedure

9.4. SUMMARY

This chapter has been dedicated to metaheuristic techniques in general and genetic algorithms in particular. Metaheuristic techniques are being successfully applied to a wide variety of combinatorial optimization problems, difficult problems, for which exact techniques do not allow us to find the solution in many cases due to the enormous computational effort required. For many of these problems, heuristic techniques that provide good results have been developed. However, metaheuristics improve these results since they make a deep search of the solution space with acceptable computational effort.

Genetic algorithms are based on the mechanisms of natural evolution, guided by the principle of survival of the fittest. Starting with an initial population of solutions, mechanisms that mimic natural processes affecting the species are applied to it, thus evolving populations of a better quality. The main difference between genetic algorithms and other metaheuristics such as tabu search or simulated annealing, is that the former handles a set or population of solutions at each iteration and the latter ones a single solution per iteration. These three techniques are the most used metaheuristics, but there is not one that can always offer the best results for any problem.

9.5. SELECTED REFERENCES

1. Alcaraz, J. (2001), *Genetic Algorithms for Resource-Constrained Project Scheduling*, ProQuest Information and Learning.

2. Alcaraz, J. and Maroto, C. (2006), A hybrid genetic algorithm based on intelligent encoding for project scheduling, en: *Perspectives in modern project scheduling,* Springer.

3. Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley.

4. Glover, F. y Laguna, M. (1997), *Tabu Search,* Kluwer Academic Publishers.

5. Holland, H.J. (1975), *Adaptation in natural and artificial systems*, University of Michigan Press.

6. Rayward-Smith, V.J., Osman, I.H., Reeves, C.R. and Smith, G.D. (Editors) (1996), *Modern Heuristic Search Methods,* John Wiley and Sons.

7. Voβ, S., Martello, S., Osman, I.H. y Roucairol, C. (Editores) (1999), *Meta-heuristics. Advances and Trends in Local Search Paradigms for Optimization,* Kluwer Academic Publishers.

8. Winston (2011), Microsoft Excel 2010: Data Analysis and Business Modeling, Microsoft Press.

9.6. CASE STUDIES

CASE STUDY 1

Solve the travelling salesman model from section 9.1.6 eliminating the cities of Zamora and Cádiz and including the city of Badajoz.

1. What is the minimum total distance travelled?
2. How long did Solver take to solve the problem?
3. Does the solution provided by Solver change if we limit the maximum number of subproblems solved to 200?
4. What is the route to follow if we want to start and finish in Madrid? And what if the path should start and end in Zamora?

CASE STUDY 2: SCHEDULING JOBS WITH DUE DATES WITH EXCEL EVOLUTIONARY SOLVER

Among the combinatorial optimization problems in which metaheuristic techniques are the only alternatives in practice, we can mention the job scheduling or sequencing operations. In general, we have to decide the order in which a series of tasks or operations are performed in order to optimize a certain criterion. The following case presents a simple example to be solved with Evolutionary Solver.

We have a set of 16 jobs that need to be performed sequentially and we know the processing time in days, and the delivery date in days from the date on which the execution of the first job starts for each of them. These delivery or due dates are approximate in the sense that they can be exceeded, but we must also try to ensure that the amount of time by which a job exceeds the delivery date is not too long.

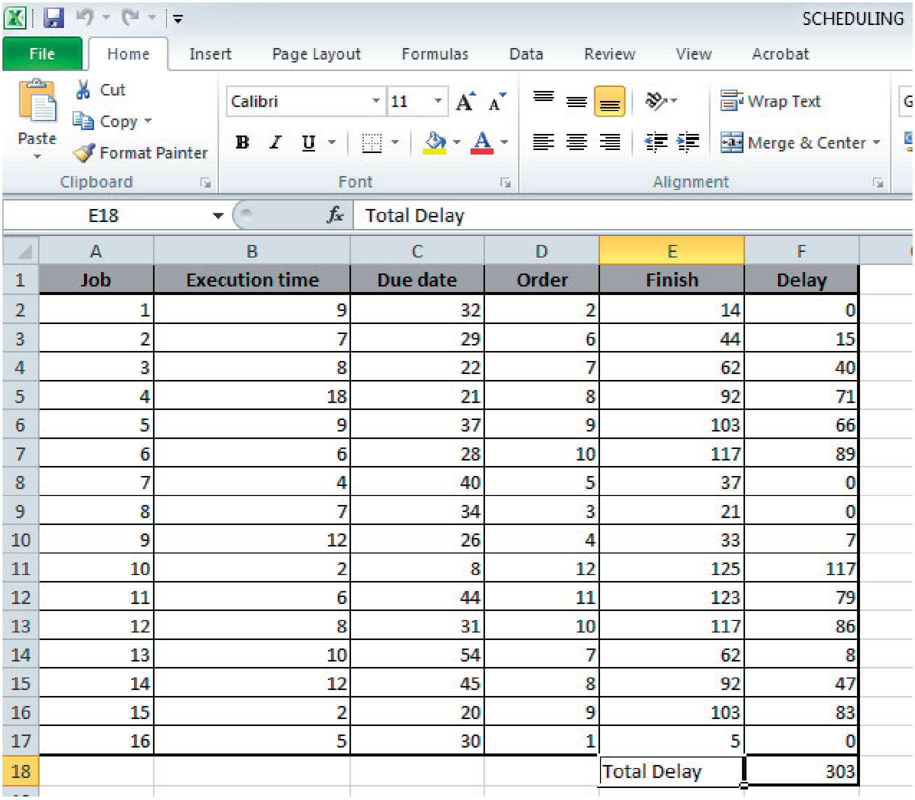
In short, the problem consists of determining the order in which jobs are to be executed so as to minimize the total number of days that a delay is incurred in terms of due dates.

Figure 9.15 shows an Excel spreadsheet with the problem data. The jobs have been numbered from 1 to 16. The execution times and due dates, in days, are presented in columns B and C.

In the previous sheet we have included three additional columns. The column *Order* is reserved to reflect the solution provided by Solver, ie, it must contain a permutation of jobs. In column *Finish*, the number of days until the end of the execution of a job, if executed in the order shown in column D, must be calculated. So, if, as is now shown in that column, the first scheduled job were job 16, with a duration of 5 days, and the second were job 1, with a duration of nine days, this work would finish after 14 days. In the last column we calculate the delay for each job, with respect to the due date, if executed in the specified order. If job 1 finishes in 14 days, the delay is 0. Finally, in cell F18 we calculate the sum of days overdue, which would be 303 in this case.

Solve the above problem with Evolutionary Solver and answer the following questions:

1. In what order should the jobs be sequenced?
2. When does job 12 finish? What day should the execution of job number 4 start?
3. How many days will job number 8 be delayed with respect to its due date?
4. What is the sum of the days delayed?
5. Which is the job with the longest delay?



	A	B	C	D	E	F
1	Job	Execution time	Due date	Order	Finish	Delay
2	1	9	32	2	14	0
3	2	7	29	6	44	15
4	3	8	22	7	62	40
5	4	18	21	8	92	71
6	5	9	37	9	103	66
7	6	6	28	10	117	89
8	7	4	40	5	37	0
9	8	7	34	3	21	0
10	9	12	26	4	33	7
11	10	2	8	12	125	117
12	11	6	44	11	123	79
13	12	8	31	10	117	86
14	13	10	54	7	62	8
15	14	12	45	8	92	47
16	15	2	20	9	103	83
17	16	5	30	1	5	0
18					Total Delay	303

Figure 9.15. Defining the job scheduling problem with Excel

ANNEX 1

THE SOLVER OF THE EXCEL SPREADSHEET

A1.1. FORMULATING AN OPTIMIZATION MODEL

Spreadsheets are data analysis tools most commonly used in the business environment. Among its benefits for improving decisions it is the possibility of solving optimization models -linear, integer and nonlinear programming models- with ***Solver*** tool. To access Solver it is necessary to install it first. For Excel Spreadsheet 2010 and 2013 you should go to Office button, choose Excel *Options*, and in the *Add-Ins* dialogue box, check whether *Solver* is installed or not. If it has not been installed, select *Excel Add-Ins* in the *Manage* window and click the *Go* button. This brings up a window with the add-ins that incorporate by default Excel and the ones you want to install can be selected. To install *Solver* we mark it and click the *OK* button. Once installed, you can access the *Solver* on the *Data* tab.

The *Solver* tool available in Excel 2010 and 1013 introduces three methods of solving optimization models, Simplex LP, GRG Nonlinear and Evolutionary. We will use the example of energy production and pollution control described in Chapter 2 to explain the input of a linear programming model in the spreadsheet and its resolution by Simplex LP method. Chapter 9 describes how to use the Evolutionary method to solve the Traveling Salesman Problem (TSP).

We always start entering the problem data on a sheet. The data of the problem we want to solve have been introduced as we can see in Figure A1.1. The **data cells** are shaded in light yellow. To facilitate the building and interpretation of the model it is convenient to use **range names**. So in this case we used SteamProduction as a range names for cells E4:F4, RHS for I6:I9 and TechnicalCoefficients is the name of the cell range E6:F9. To enter a range name we just have to select the cells and go to *Formulas/Define name*. Range names can not contain spaces. So if you have more than one word, it is useful to begin each word with a capital letter and eliminate spaces to facilitate understanding.

As we have already seen, in Chapters 1 and 2, to correctly formulate an optimization model in general and a linear programming model in particular, we need to know:

1. The decisions to make.

2. The constraints we have.

3. How to measure the performance of our decisions.

In this case the **decisions** are tons of each type of coal that we will use to produce steam, which is transformed into energy, i.e. the model **variables**. **Constraints** that limit the values that can take these decision variables are the firm's technological capacity (capacity of the loading system and the pulverizer) and environmental regulatory restrictions limiting the release of pollutants (smoke and sulfur oxide). The more energy the company can produce with available resources and meeting environmental regulations the better our decisions. Therefore, the **objective** is to maximize the total steam production.

Table A.1.1. Data of the energy production and pollution control problem (Chapter 2)

	A	B	C	D	E	F	G		I
1	**ENERGY PRODUCTION AND POLLUTION CONTROL**								
2									
3					**Coal A**	**Coal B**			
4	Steam production in thousands of lb/ton				24	20			
5									**RHS**
6	Emission of smoke kg/h				0,5	1			12
7	Loading installation				1	1			20
8	Pulverizer capacity				1,5	1			24
9	Emission of sulphur				1200	-800			0
10									

The decision variables will appear in cells E12:F12 to which we assigned the range name UsedCoal. The cells where *Solver* must store the value of the variables of the solved model should be reserved for that purpose. They are the variable cells, which may initially be assigned a zero value and after solving the model we will have the optimal variable values. The constraint values are introduced in CapacityUtilization range in the cells G6:G9. These cells collect the first member of the constraints of the problem (Chapter 2, section 2.2.3). As shown in Figure A.1.2 it is calculated:

G6= =SUMPRODUCT(E6:F6;UsedCoal)

G7=SUMPRODUCT(E7:F7;UsedCoal)

G8=SUMPRODUCT(E8:F8;UsedCoal)

G9=SUMPRODUCT(E9:F9;UsedCoal)

For this small example, we could have directly used the product and the sum. However, with the SUMPRODUCT function of the spreadsheet we illustrate how to introduce constraints of linear models. These cells represent the value of the left hand side (LHS) of the constraints in the optimal solution. Therefore, they are **result cells** which contain the values of the left hand of the constraints, i.e. they depend on the value that the decision variables have taken.

In cells H6:H9 we input the signs of the constraints to facilitate understanding the data in the spreadsheet. To solve the model these conditions of lower, higher or equal operators must be specified in a dialogue box as we will see in section A1.2.

Finally we have to enable a cell to calculate the value of the objective function. This is what we have done in cell I12 using the SUMPRODUCT function. All this information can be seen in Table A1.2.

I12=MAX (SUMPRODUCT (SteamProduction;UsedCoal))

Table A.1.2. Model for the energy production and pollution control problem (Chapter 2)

	A	B	C	D	E	F	G	H	I
1	**ENERGY PRODUCTION AND POLLUTION CONTROL**								
2									
3					**Carbón A**	**Carbón B**			
4	Steam production in thousands of lb/ton				24	20			
5							**Used capacity LHS**		**RHS**
6	Emission of smoke kg/h				0,5	1	0	≤	12
7	Loading installation				1	1	0	≤	20
8	Pulverizer capacity				1,5	1	0	≤	24
9	Emission of sulphur				1200	-800	0	≥	0
10									
11					**Coal A ton/h**	**Coal B ton/h**			**Total Steam Production thousand lb/h**
12					0	0			0

We must emphasize the usefulness of the SUMPRODUCT function of the spreadsheet to input linear functions representing the objective function and constraints in linear programming models. Also when using range names in formulas we simplify the data entry process and improve the understanding of the model. When the coefficients multiplied by the decision variables are all 1, we can obviously use the SUM function.

It is known that the *relative references* in formulas refer to cells because of its position in relation to the cell containing the formula. *Absolute references* refer to cells by their fixed position in the spreadsheet. When we copy and paste the relative references they are automatically adjusted while absolute references are not. The ones which include range names are treated as absolute references. Another way to indicate an absolute reference is to use F4 or insert a $ sign before the letter and number of the cell.

Table A.1.2 shows the complete formulation of the model of the energy production. To sum up, we first introduce the problem **data** in the data cells, which correspond to the **technical coefficients (a_{ij})**, **the objective function coefficients (C_j)** and the **right hand sides (RHS) of the constraints (b_i)** of the model. After that we define the **variable cells** in which we have the **decision variable** values. Finally, we introduce the linear functions that represent the **constraints** and the **objective function**. Then the model can be solved. Now, after entering the data of the problem, which may have been made in other ways, we must tell *Solver* how to build the linear programming model to solve on these data.

A1.2. SOLVING A LINEAR PROGRAMMING MODEL

As explained in the previous section the *Solver* tool allows us to solve optimization models, linear, integer and non-linear programming. Let us look at how to solve the energy production model with *Simplex LP* method, specific to solve linear programming or integer linear programming models.

In order to solve the model with *Solver* we need to specify where it is in the spreadsheet. First, we double click on *Solver* and the dialogue box shown in Figure A.1.1 appears.

We can select the objective cell and the decision variables simply by selecting them in the spreadsheet. The constraints are introduced with the *Add* button, which allows us to select the cell containing the LSH of the constraint, the sign (≥, ≤ or =) and the right hand side (RHS).

In the drop down list that lets us select whether the constraint is ≥, ≤ or =, ***int*** and ***bin*** also appear to indicate whether the variables are integer or binary in the integer programming models. There also appears *dif*, which may be imposed on a set of variables and should require that the set of values of the variables is composed of a permutation of integers, all different, which vary between 1 and the number of variables listed. Several examples that make use of this option can be found in Chapter 9.

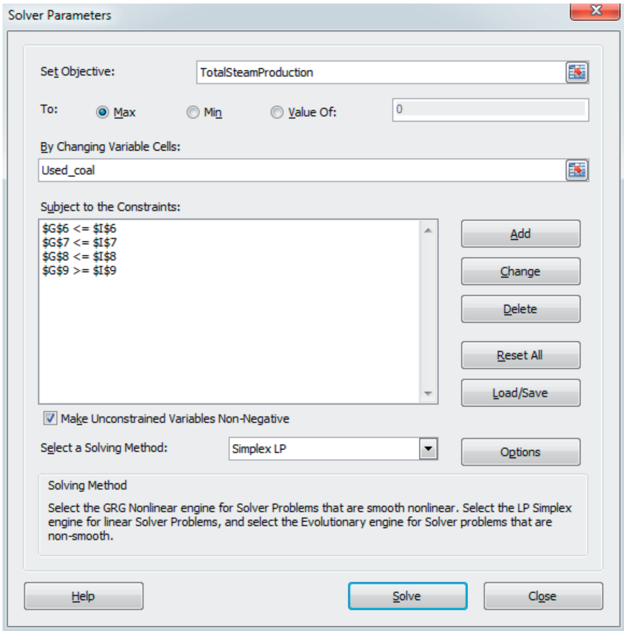


Figure A.1.1. *Solver* dialogue box: specifying the model

In the dropdown that appears in the option *Select a Solving Method* there are three possibilities:

- Simplex LP: to solve models in which both the objective function and the constraints are linear.
- GRG Nonlinear: to solve smooth models in which there are nonlinear functions.
- Evolutionary: to solve non-smooth models.

We chose the *Simplex LP* method because it is a linear programming model. In this case *Solver* applies the Simplex algorithm to find the optimal solution. By checking the box *Make Unconstrained Variables Non-Negative*, we impose the non-negativity condition for the model variables. The model is ready to be solved. However, before solving it with the method chosen, it is convenient to fix some of the options available. To do this, click the *Options* button, available in the *Solver* dialogue box, shown in Figure A.1.2. As it can be seen in this figure, there are three different tabs to set options, depending on the method that we will use to solve the model. The first tab, *All Methods*, provides options that affect all three methods.

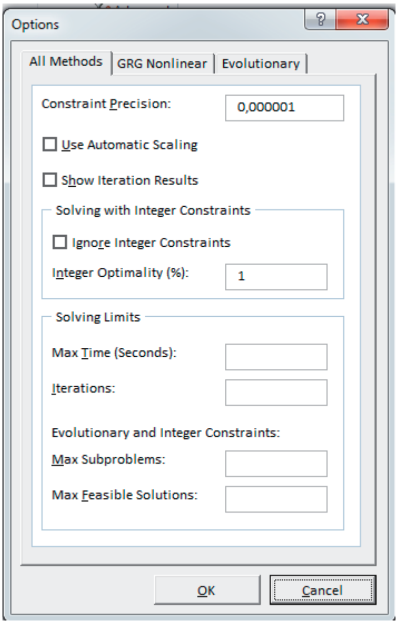


Figure A.1.2. *Solver* options dialogue box

The options that appear on that tab when we are using the *Simplex LP* method are useful, mainly when integer variables appear in the model (or binary, as the binary is a particular case of integer). In that case the resolution time of the problem can be quite high and we can reduce it by setting values for any of these parameters. In many cases, limiting the computation time can thereby mean forcing the method to finish before finding the optimal solution. Since, in this case, we have a simple linear programming model in which there are no integer variables, options can be left at the default values shown, so that the *Solver* will invest the time needed to solve it and the solution found will be optimal, if there is at least one. However, let us comment, the meaning of the parameters that can be set in this tab:

- **Constraint Precision**

A constraint is satisfied when it is true within a small tolerance, which can be specified in this option.

- **Use Authomatic Scaling**

Select this option to apply a scaling process to the model introduced when there are wide differences in magnitude between the inputs and outputs, for example, when maximizing the percentage of benefits (the target value will be between 0 and 1, or 0 and 100) based on investments of millions of dollars.

- **Show Iterations Results**

Select this option to make *Solver* display the results of each iteration.

- **Ignore Integer Constraints**

If you check this box *Solver* solves the problem by assuming that all variables are continuous.

- **Integer Optimality**

The Integer Optimality % default setting is a compromise value that often saves a great deal of time, and ensures that a solution found by solver is within 1% of the true optimal solution. We can change this percentage. If you want to find the optimal solution set the Integer Optimality % tolerance to zero.

Max. Time (seconds)

Limits the time the solution process takes, so if the set time is reached, the solution provided by *Solver* when solving a linear programming model may not be optimal.

- **Iterations**

Limits the number of iterations that can carry out the solution process, so if the limit is reached, the solution provided by *Solver* to solve a linear programming model may not be optimal.

- **Max. Subproblems**

Limits the number of subproblems that can be solved by the resolution process. When solving a linear integer programming model, *Solver* applies an advanced version of the branch and bound algorithm studied in Chapter 5, limiting with this parameter the number of subproblems solved.

- **Max. Feasible Solutions**

Setting this parameter, *Solver* limits the number of feasible solutions that can explore the process before completion.

Once you set the options according to your needs, click the *OK* button and then the *Solve* button in the *Solver* dialogue box. Having found the optimal solution, the program displays the screen, shown in Figure A.1.3, with the three types of reports that can be generated. Select *Answer* and *Sensitivity* reports, *Solver* generates those reports in the spreadsheet and furthermore, it completes in the sheet of the model the formulas that depend on the value found for the variables, as can be seen in Table A.1.3, where we can check the optimum value of the decision variables (12 ton/h of coal A and 6 ton/h of coal B) and the value that maximizes the objective function (408 thousand pounds per hour).

In Tables A.1.4 and A.1.5 we can find the *Answer* and *Sensitivity* reports. Table A.1.3 shows the model and the optimal solution. That is, the values of the decision variables.

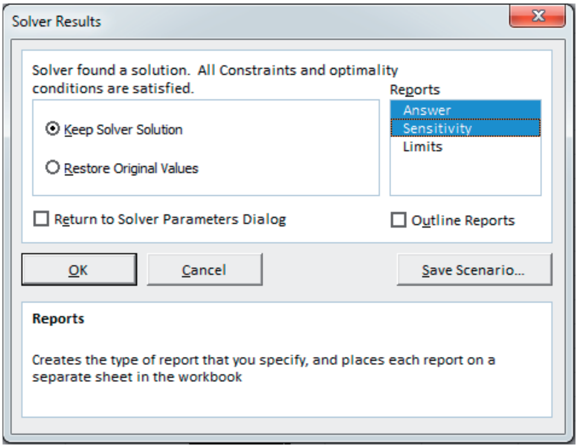


Figure A.1.3. *Solver* results dialogue box

Table A.1.3. Optimal solution for the energy production and pollution control problem (Chapter 2)

	A	B	C	D	E	F	G	H	I
1	**ENERGY PRODUCTION AND POLLUTION CONTROL**								
2									
3					**Coal A**	**Coal B**			
4	Steam production in thousands of lb/ton				24	20			
5							**Used capacity LHS**		**RHS**
6	Emission of smoke kg/h				0,5	1	12	≤	12
7	Loading installation				1	1	18	≤	20
8	Pulverizer capacity				1,5	1	24	≤	24
9	Emission of sulphur				1200	-800	9600	≥	0
10									
11					**Coal A ton/h**	**Coal B ton/h**			**Total Steam Production thousand lb/h**
12					12	6			408

Table A.1.4. ***Solver Answer*** **report for the energy production and pollution control problem (Chapter 2)**

Microsoft Excel 14.0 Answer Report

Worksheet: [SteamProduction.xls]Data1

Report Created:

Result: Solver found a solution. All Constraints and optimality conditions are satisfied.

Solver Engine

Engine: Simplex LP

Solution Time: 0,016 Seconds.

Iterations: 3 Subproblems: 0

Solver Options

Max Time Unlimited, Iterations Unlimited, Precision 0,000001

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume NonNegative

Objective Cell (Max)

Cell	Name	Original Value	Final Value
$I$12	TotalSteamProduction	0	408

Variable Cells

Cell	Name	Original Value	Final Value	Integer
$E$12	Coal A (ton/h)	0	12	Contin
$F$12	Coal B (ton/h)	0	6	Contin

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$G$6	Emission of smoke kg/h	12	$G$6<=$I$6	Binding	0
$G$7	Loading installation	18	$G$7<=$I$7	Not Binding	2
$G$8	Pulverizer capacity	24	$G$8<=$I$8	Binding	0
$G$9	Emission of sulphur oxide	9600	$G$9>=$I$9	Not Binding	9600

Table A.1.4 presents the *Answer* report, which has three sections, which refer to the objective function, variables and constraints. The first section shows the value of the objective function, which in our problem is 408, which means that the maximum steam production is at 408,000 pounds of steam per hour. The second section displays the optimal solution, namely the optimum value for each of the variables, and the nature of these (continuous in this case).

Finally, for each of the constraints of the model it first indicates the value taken by the left hand side of the constraint. The status indicates whether the constraint is strictly satisfied (binding) or not (non-binding). The last column indicates the slack of that constraint.

Sensitivity report (Table A.1.5) presents information relating to sensitivity analysis. For each variable the report indicates the value taken in the optimal solution, the reduced cost (the quantity that should improve the coefficient of the variable, if its value is zero, so it could take a nonzero value in the optimal solution) and the possible increase and decrease of the coefficient of the variable without changing the optimal solution. Regarding the constraints *Shadow Price* gives its opportunity cost and the interval in which its right hand side may differ from the initial value so that the opportunity cost remains constant.

Table A.1.5. *Solver* Sensitivity report for the energy production and pollution control problem (Chapter 2)

Microsoft Excel 14.0 Sensitivity Report
Worksheet: [SteamProduction.xls]Data1

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$E$12	Coal A (ton/h)	12	0	24	6	14
$F$12	Coal B (ton/h)	6	0	20	28	4

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$G$6	Emission of smoke kg/h	12	6	12	4	4
$G$7	Loading installation	18	0	20	1E+30	2
$G$8	Pulverizer capacity	24	14	24	4	6
$G$9	Emission of sulphur oxide	9600	0	0	9600	1E+30

A1.3. SOLVING OTHER TYPES OF MODELS

As already discussed above *Solver* also incorporates, apart from Simplex LP, that solves any linear programming model (continuous or integer), two other techniques, GRG Nonlinear and Evolutionary that allow us to solve nonlinear models, whether smooth or non-smooth.

Nonlinear models are those in which the objective functions and/or any of the constraints contains references to formulas that do not follow the pattern variables multiplied by constants. If, for example, x and y are variables of the model, any appearance of references to the same formulas as below would make the model nonlinear:

- x^2
- xy
- $\sin x$
- x^y

If the model only includes functions with ordinary mathematical operators as above, we can use the GRG Nonlinear method to solve it. That procedure uses the generalized reduced gradient method, based on finding points where the slope is zero, the condition met by the maxima and minima. In general, methods for solving nonlinear models start searching at a given starting point and they approach the solution in successive iterations, so that the solution found may depend on the chosen starting point, particularly in functions that may have different local maxima or minima. To avoid this inconvenience, GRG Nonlinear features a *Multiple Start* option, which allows us to specify the number of starting points. The process solves the problem starting from each of these points and eventually returns the best solution found. This option works best if variables are imposed reasonable upper and lower bounds.

The *GRG Nonlinear* method is not suitable when in the model references including non-smooth functions appear, such as MAX, MIN, ABS, IF, SUMIF, SUMIFS, COUNTIF, COUNTIFS. In this case, we recommend using *Evolutionary* method, based on genetic algorithms, described in Chapter 9. The options of this method allows us to, among other things, specify the size of the population to be used, set the mutation rate or limit the time that can elapse without improving the value of the objective function to complete the search process. In Chapter 9 it is described how to use this method to solve two optimization problems; the traveling salesman problem and a problem of sequencing jobs.

A1.4. BUILDING GOOD SPREADSHEET MODELS

There are many ways of representing a model in a spreadsheet and one of the advantages is precisely the flexibility it offers. Although Excel has many features such as range names, shadows, borders... that create "good" models that are easy to understand, debug and modify, it is also easy to create bad models. Here are some tips that will facilitate the construction of good models.

1. Enter, organize and clearly identify the data

The full model is built on the data structure. We must carefully introduce and present all data before the rest of the model. The structure of the model should fit the data as much as possible.

We should group the data conveniently and put labels that clearly identify them. In the data presented in a table we should put headers with an overview and each column and each row should have the name that identifies the input data. We must also identify data units. Enter data oriented in the same way is not only clearer, but also allows us to use the SUMPRODUCT function. This function assumes that the two ranges have exactly the same number of rows and columns.

2. Enter each data only in a single cell, do not repeat data in different cells

If data is needed in more than one formula, always refer to the original data cell instead of repeating the data in different sites. This model could be modified more easily. If the data change, we would only need to modify them once. We would not need to search in the whole model how many times the data that have changed appear.

3. Separate data from formulas. The formulas should refer to the data cells

Avoid using data directly in formulas. You have to enter the numbers in the data cells and refer to them where necessary. Separating data from the formulas has a double advantage. Firstly, all data are visible in the spreadsheet instead of being hidden in the formulas. To view the data makes the model easier to interpret. Secondly, the model is easier to modify because changing data only requires modifying the corresponding data cell. We do not need to modify formulas. This is important in the sensitivity analysis.

4. Keep the model simple and as easy to interpret as possible

Avoid using complicated functions of Excel when there are available functions that are simpler and easier to interpret. Use whenever possible SUMPRODUCT or SUM. This makes the model easier to interpret and helps to ensure that the model is linear (linear models are much easier to solve than nonlinear).

5. Use range names

One way to refer to a group of cells or a cell in a formula in the spreadsheet is to identify it with a letter and number. A better alternative is to use descriptive range names. To do this, select the cells and put the range name. The range names are especially important when writing a formula for a cell result. Writing the formula in terms of range names makes it easier to interpret. Range names also make that the description in the *Solver* model is easier to understand. In general, it is advisable to name a range for each group of data cells, variable cells, objective cell and the two sides of the constraints, LHS and RHS.

As no spaces are allowed in range names, we should start each word with a capital letter to enhance understanding. When we modify a model using range names we have to ensure that the range names still refer to the correct cells. When rows or columns are inserted, it must be done in the middle of the range, and not at the end.

6. Using absolute and relative references to copy formulas easily

When we need to use a formula several times, we can introduce it once and then use Excel commands to replicate. Using absolute and relative references in the formula not only helps building models, but also makes them easier to change.

7. Use borders, shading and colours for different types of data

It is very important to distinguish the data cells, variable cells, the result cells and the objective cell in the spreadsheet. The use of shadows, borders and colours help us visualize the model quickly.

8. View the entire model in the spreadsheet

The *Solver* uses a combination of spreadsheet and the *Solver* dialogue box to specify the model to be solved. For example, we can specify the inequalities in the *Solver* dialogue box without putting them into the spreadsheet. However, it is recommended that each model element appears on the screen. This is useful for later updates. In particular, all the elements of a constraint must appear on the screen. A good test is not using the *Solver* dialogue box in order to understand any model element. We must be able to identify the variable cells, the objective cell and constraints only by looking at the spreadsheet.

ANNEX 2

THE MODELLING LANGUAGE AND OPTIMIZER: LINGO

A modelling language and an optimizer are necessary tools to solve real decision making problems in practice. Thus they are also essential in teaching and learning the techniques that allow us to formulate, model and solve real decision making problems. Among software packages available in the market we have chosen LINGO for several reasons. First, it is an optimization software that incorporates a modelling language which allows generating big optimization models easily. It also permits importing and exporting data from and to Excel and databases. Furthermore it is available for Linux platform and a wide variety of hardware systems (compatible PC, Macintosh and working stations). As well as this for different sizes of models, from student's versions that may solve problems with 200 variables and 100 constraints, to more powerful versions that allow solving models with an unlimited number of variables and constraints. Evaluation versions can be downloaded from the LINDO Systems website (www.lindo.com).

A2.1. FEATURES OF LINGO

LINGO is an optimizer that allows us:

1. To solve direct models consisting of equations with independent variables and simultaneous equation systems.
2. **To solve optimization models** in which an objective function has to be maximized or minimized and whose variables must fulfil a number of constraints. It solves linear, integer, nonlinear and stochastic programming models. A ***Global*** optimizer is also included to find a global optimum in nonlinear programming. In addition a new feature is *Chance-Constrained Programming*. In this case one or more sets of constraints are allowed to be violated with a specified probability. This tool is useful when certain resources or demands are random.

3. To generate models through a modelling language, particularly useful for large models with many equations of similar structure. Furthermore, it allows the creation of the structure of the model and keeping it apart from the model's data. Data can be loaded from a file or spreadsheet.

When using the modelling language, the system converts the expressions to the required form to be solved with the appropriate algorithm. For **linear programming** models the system uses the **revised simplex method**. It also has the interior point algorithm (*Barrier Solver)*, useful for solving linear and quadratic models with a very big number of variables and constraints. If the system detects **integer variables**, it finds the solution with the **branch and bound algorithm**, adding **cuts** to limit the non-integer feasible region. For nonlinear models, it uses the generalized reduced gradient algorithm and the sequential linear programming algorithm. A ***Global*** optimizer is also included to find a global optimum in nonlinear programming. Finally, LINGO also incorporates a library with statistical, financial and mathematical formulae.

A2.2. ENTERING AND SOLVING MODELS

The models can be input in the conventional way typing the functions or using the modelling language. In both cases the model is between two commands which are MODEL: and END.

For example, the model

$Max\ 24\ X_1 + 20\ X_2$

$0.5\ X_1 + X_2 \leq 1$

$X_1 + X_2 \leq 20$

$1.5\ X_1 + X_2 \leq 24$

$1200\ X_1 - 800\ X_2 \geq 0$

$X_1 \geq 0 \quad X_2 \geq 0$

is indicated in the following way:

```
MODEL:
!EXAMPLE 1: ENERGY PRODUCTION AND POLLUTION CONTROL;
[OBJ] MAX = 24 * X1 + 20 * X2;
[SMOKE] 0.5 * X1 + X2 <= 12;
[LOAD] X1 + X2 <= 20;
[PULVERIZER] 1.5 * X1 + X2 <= 24;
[SULPHUR] 1200 * X1   800 * X2 >= 0;
END
```

Note that in the input data we have to type the symbol ***** for **multiplication** and semicolon (**;**) for **the end** of a sentence, which may be a **comment, the objective function** or **constraints**. The **comments** start with the exclamation mark (**!**) and the names of the objective function and the constraints in brackets. We can also write variables in the Right-Hand-Side (RHS) of constraints.

In LINGO menu, the option ***Solve*** displays the optimal solution, and the option ***Range*** the sensitivity analysis, as shown below for the previous example. To obtain the sensitivity analysis, it is necessary to select Prices & Ranges and the model has to be solved previously. To do so go to ***LINGO menu / Options…/ General Solver/ Dual computations/ Prices & Ranges.***

```
Global optimal solution found at step:          5
Objective value:                         408.0000

         Variable            Value        Reduced Cost
               X1         12.00000           0.0000000
               X2          6.00000           0.0000000

              Row    Slack or Surplus        Dual Price
              OBJ         408.0000            1.000000
            SMOKE           0.0000            6.000000
             LOAD           2.0000            0.000000
       PULVERIZER           0.0000           14.000000
          SULPHUR        9600.0000            0.000000

Ranges in which the basis is unchanged:

                      Objective Coefficient Ranges
                 Current          Allowable         Allowable
Variable       Coefficient         Increase          Decrease
      X1         24.00000         6.000000          14.00000
      X2         20.00000        28.000000           4.00000

                       Right hand Side Ranges
         Row     Current          Allowable         Allowable
                     RHS           Increase          Decrease
       SMOKE     12.00000          4.000000          4.000000
        LOAD     20.00000          INFINITY          2.000000
  PULVERIZER     24.00000          4.000000          6.000000
     SULPHUR      0.00000       9600.000000          INFINITY
```

After solving the model we first obtain general information about the number and type of variables, constraints and coefficients. Then we can see the **optimal value of the objective function,** in this example 408. For each **variable** it indicates the activity level in the optimal solution (X1 =12 and X2= 6) and the **reduced cost**. The reduced cost may be interpreted as the amount that the coefficient of the objective function of that variable must improve for it to take a value other than zero in the optimal solution. When the variable has a positive value due to a lower bound higher than zero, the reduced cost is the penalty of objective function for increasing an additional unit of the variable in the solution.

The solution report provides the value of the **slack variables** (***slack or surplus***) of the constraints and their **opportunity cost or *dual price***. Note that the first row is not a constraint, but the objective function of the model. The dual price is the amount that the **objective function improves per increasing unit in the RHS of the constraint**.

A2.3. MODELLING LANGUAGE

3.1. DEFINITION OF SETS

In large models it is frequent to have similar sets of variables and constraints. The use of "sets" allows us to define and work with classes of objects that have to be processed in a similar way. Sets are the basis of LINGO's modeling language through which we can write sets of similar constraints in a sentence and express long formulations in a more compact form.

The sets represent groups of similar objects. One set can be, for example, a list of products or tasks. Each element of the set may have one or more characteristics, called attributes. The attributes may be formed by known or unknown data. Thus a set of products may have an attribute such as the price of each product.

LINGO recognizes two types of sets: primitive and derived. For example, in a transport model, the set formed by three factories is a primitive set, like the set formed by four warehouses. A derived set may be created from one or more sets or is a subset of another set, or the combination of elements of other groups. Thus in the transport problem, a derived set is the set formed by the delivery of products from the three factories to the four warehouses.

These sets are defined in an optional section of LINGO called SETS. In this section an unlimited number of groups or sets of objects can be defined and starts with SETS: and ends with ENDSETS. The definition of a primitive set has the following syntax:

Name/ members/: attributes;

The system lets you establish a number of members without explicitly naming them, by using the form /1...n/ instead of the members' names.

3.2. MODEL DATA

It is usually necessary to input the values of some given attributes before solving the model. Similarly to the sets section, the data section starts with the word DATA: and ends with ENDDATA.

In this section the data of the attributes defined in sets are written with the following syntax:

attribute = list_of_values;

Logically, the list of values should have the same number of elements as the set to which the attribute belongs, each value being separated by spaces or commas. The only exception is when the same value is to be given to each element of the vector. In this case, only a value is supplied and LINGO assigns it to each element.

Sometimes you will only want to supply values to some of the elements of the attribute array, leaving the other elements unknown so that LINGO has to find their optimal value. In such a case, leave a space in the position of this value, placing it between commas. If it is the first value, the sentence starts with a comma, and if it is the last value of the array, it ends with a comma. Observe that although LINGO accepts blank spaces or commas as delimiting marks for data sentences, the commas should be used to delimit unknown data.

We may want to supply the values of some or all the elements of an attribute each time we solve the model. LINGO accepts the question mark instead of the value that we have to specify. Later, LINGO will ask for these parameters; only numerical values are valid as an answer.

Furthermore, in nonlinear models an optional section that starts with INIT: and ends with ENDINIT can be used.

INIT:

attribute = list_of_values;

ENDINIT

This section serves to set the initial value of the variables. These data are used as an initial point for the algorithm. The list of values should contain as many data as defined in the corresponding attribute, except when we want it to have the same value for all the elements, indicated only once. In this section, as in the data section, some values can be left unspecified typing two commas or starting/ending and a comma if it is the first/last element of the array. Similarly, question marks (?) can be used to specify some given values when solving the model.

3.3. FUNCTIONS

Function @FOR allows us to perform operations with the elements of the sets, whose syntax is:

@FOR (name_set (list_of_índices_of_ set) | conditional qualifier: expression);

The conditional qualifier is optional and if used, it should be preceded by the character |. This function requires at least one expression, but an unlimited number of expressions separated by semicolons can be used (;).

The list_of_índices_of_ set is optional. If not used, LINGO applies the expression to all members of the set. When specified, the indices of the list can be used in the conditional qualifier.

Let us see an example to calculate 1/X from a list of numbers.

```
MODEL:
SETS:
NUM/ 1.. 10/ : VAL, REC;
ENDSETS
@FOR (NUM(I)| VAL(I) #NE# 0: REC(I) = 1/ VAL(I) );
@FOR (NUM(I)| VAL(I) #EQ# 0: REC(I) = 0);
DATA:
VAL = 2, 0, 8, 40, 1, 0, 0.33, 50, 3, 0.2;
ENDDATA
END
```

Other functions that can be applied to sets in a similar way as @FOR, are @SUM, @MIN and @MAX. There are also auxiliary functions that allow us to work with sets, like @IN, @INDEX, @WRAP and @SIZE.

The derived sets are sets obtained from other sets. Derived sets can be defined in three ways:

1. Each derived set is complete and defined using this format

name_set (set_origin)[: attributes];

Example: delivery (factory, warehouse): cost, quantity;

2. The derived set is little dense or disperse and the members are specified by

name_set (sets_origin) / explicit_list / [: attributes];

For example, a company has two plants (A, B) and manufactures 3 products (X, Y and Z). In the first plant it produces X and Y and in the second Y and Z.

PLPR (PLANTA, PROD)/A, X, A, Y B, Y B, Z/: COSTE_UNITARIO;

3. The set is little dense and we specify the requirements that the members must fulfill as a logical expression using the following format.

name_set (sets_origin) ½ condition [: attributes];

In this case, each potential member of the set must satisfy the condition specified to become a member of the derived set

Example: A transport problem

```
 MODEL:
 !Transport problem with 3 factories and 4 warehouses;
SETS:
  FACTORY/ F1, F2, F3/: AVAILABILITY;
  WAREHOUSE / A1, A2, A3, A4/: DEMAND;
  DELIVERY(FACTORY,WAREHOUSE): COST, X;
ENDSETS
!The variables are the amounts sent from each factory I to each
warehouse J (XIJ)
Objective function: Minimize total cost;
[OBJ] MIN = @SUM(ENVIO: COSTE * X);
! Constraints
DEMAND;
   @FOR( WAREHOUSE(J): [RDEMAND]
         @SUM( FACTORY (I): X (I,J)) >= DEMAND (J));
! OFFER;
   @FOR(FACTORY(I): [ROFFER]
         @SUM(WAREHOUSE (J): X (I,J)) <= AVAILABILITY(I));
! Parameters of the model;
   DATA:
         AVAILABILITY = 30, 25, 21;
         DEMAND = 15, 17, 22, 12;
               COST = 6, 2, 6, 7,
                        4, 9, 5, 3,
                       8, 8, 1, 5;
   ENDDATA
END
```

As you can see in the previous example, in a model you can distinguish three sections:

1) The definition of the variables and sets.

2) The equations of the model.

3) The data of the model.

The algebraic format generated with LINGO and the solution are:

```
MIN 6 X( F1, A1) + 2 X( F1, A2) + 6 X( F1, A3) + 7 X( F1, A4) + 4 X( F2,
A1) + 9 X( F2, A2) + 5 X( F2, A3) + 3 X( F2, A4) + 8 X( F3, A1) +
8 X( F3, A2) + X( F3, A3) + 5 X( F3, A4)


   SUBJECT TO
   RDEMAND(A1)]  X(F1,A1)+X(F2,A1)+X(F3,A1)>=15
   RDEMAND(A2)]  X(F1,A2)+X(F2,A2)+X(F3,A2)>=17
   RDEMAND(A3)]  X(F1,A3)+X(F2,A3)+X(F3,A3)>=22
   RDEMAND(A4)]  X(F1,A4)+X(F2,A4)+X(F3,A4)>=12
   ROFFERT(F1)]  X(F1,A1)+X(F1,A2)+X(F1,A3)+X(F1,A4)<=30
   ROFFERT(F2)]  X(F2,A1)+X(F2,A2)+X(F2,A3)+X(F2,A4)<=25
   ROFFERT(F3)]  X(F3,A1)+X(F3,A2)+X(F3,A3)+X(F3,A4)<=21
   END
```

```
Global optimal solution found at step:            7
Objective value:                           161.0000

Variable                       Value        Reduced Cost
AVAILABILITY( F1)         30.00000           0.0000000
AVAILABILITY( F2)         25.00000           0.0000000
AVAILABILITY( F3)         21.00000           0.0000000
DEMAND( A1)               15.00000           0.0000000
DEMAND( A2)               17.00000           0.0000000
DEMAND( A3)               22.00000           0.0000000
DEMAND( A4)               12.00000           0.0000000
COST( F1, A1)             6.000000           0.0000000
COST( F1, A2)             2.000000           0.0000000
COST( F1, A3)             6.000000           0.0000000
COST( F1, A4)             7.000000           0.0000000
COST( F2, A1)             4.000000           0.0000000
COST( F2, A2)             9.000000           0.0000000
COST( F2, A3)             5.000000           0.0000000
COST( F2, A4)             3.000000           0.0000000
COST( F3, A1)             8.000000           0.0000000
COST( F3, A2)             8.000000           0.0000000
COST( F3, A3)             1.000000           0.0000000
COST( F3, A4)             5.000000           0.0000000
X( F1, A1)                2.000000           0.0000000
X( F1, A2)                17.00000           0.0000000
X( F1, A3)                1.000000           0.0000000
X( F1, A4)                0.0000000          2.0000000
X( F2, A1)                13.00000           0.0000000
X( F2, A2)                0.0000000          9.000000
X( F2, A3)                0.0000000          1.000000
X( F2, A4)                12.00000           0.0000000
X( F3, A1)                0.0000000          7.000000
X( F3, A2)                0.0000000         11.00000
X( F3, A3)                21.00000           0.0000000
X( F3, A4)                0.0000000          5.000000

                 Row    Slack or Surplus      Dual Price
                 OBJ        161.0000            1.000000
        RDEMAND( A1)       0.0000000           -6.000000
        RDEMAND( A2)       0.0000000           -2.000000
        RDEMAND( A3)       0.0000000           -6.000000
        RDEMAND( A4)       0.0000000           -5.000000
         ROFFER( F1)        10.00000            0.000000
         ROFFER( F2)       0.0000000            2.000000
         ROFFER( F3)       0.0000000            5.000000
```

A2.4. VARIABLE DOMAIN FUNCTIONS: BOUND, FREE, INTEGER, BINARY AND SEMICONTINUOUS

By default LINGO restricts **variables** to **nonnegative** values if ***Variables assumed non-negatives*** box is checked in *LINGO* Menu*/ Options/ General Solver*. That is, the variables can assume any real value from zero to positive infinity.

@BND (lower_bound, variable_name, upper_bound) assigns lower and upper bounds to the variables. It is important to remember that a more efficient algorithm is used to solve the model if the bounds are indicated in this way. In addition, bounds do not count as constraints and larger models can be solved.

@FREE(variable_name) allows the variable to take negative values, i.e., between negative infinity and positive infinity.

@GIN (variable_name) makes the variable only take integer values.

@BIN(variable_name) makes the variable be binary, i.e., take only 0/1 values.

@SEMIC(lower_bound, variable_name, upper_bound) indicates a semicontinuous variable which is zero or lies within nonnegative range. LINGO generates the necessary binary variables and constraints that count for the size of models.

LINGO supports SOS (Special Ordered Sets) and the following types of @SOS functions:

@SOS1 At most, only one variable belonging to an SOS1 set will be greater than zero.

@SOS2 At most, only two variables in a SOS2 set can be different from zero. If two variables are nonzero, then the variables will be adjacent to one another.

@SOS3 In the SOS3 set one variable will be equal to 1 exactly and all remaining variables will be equal to zero.

Any variable in *SOS sets* count as integer variables against the limit of integer variable imposed in some versions of LINGO.

@CARD

This function is related to cardinality sets of variables. It allows specifying a set of variables with at most N variables allowed to be nonzero. This function can improve the efficiency of branch & bound algorithm and reduce the number of variables and constraints of the model.

A2.5. MENUS: FILE, EDIT, LINGO, WINDOW AND HELP

LINGO for Windows has five menus: File, Edit, LINGO, Windows and Help. We will describe some options briefly.

FILE Menu:

New (F2) creates a model.

Open (Ctrl+O) opens an existing text file.

Save (Ctrl+S) saves the active window as text. It may save models, reports or commands.

Save as (F5) saves the active window with the name given by the user in the dialog box. It can save models, reports or commands.

Close (F6) closes the active window. If it is a model with no name or the file has been modified, the program asks if you want to save the changes.

Print (F7) sends the information of the active window to a printer.

Print Setup (F8) to select a printer.

Print Preview (Shift+F8) to view the document.

Log Output (F9) sends all the following screens to a text file. You can select, overwrite on the existing file or add the following output.

Take Commands (F11) Use this option to read batch files with models and commands to execute automatic operations.

Export File (IMPORT and EXPORT). It serves to import and export files in MPS format. This format developed by IBM is useful to transfer models to other software or platforms.

Database User Imp. For entering user ID and password information to access data base with @ODBC function.

License shows information of software license.

Exit (F10) to exit LINGO.

EDIT Menu:

Undo (Ctrl+Z) undoes the last action.

Redo (Ctrl+Y) undoes the last command undo.

Cut (Ctrl+X) cuts the selected text and sends it to the clipboard.

Copy (Ctrl+C) copies the selected text to the clipboard.

Paste (Ctrl+V) inserts the selected text at the place indicated by the cursor.

Paste Special opens a dialog for pasting objects.

Select All (Ctrl+A) selects all the contents of the edition window.

Find… (Ctrl+F) searches for a string of text in the active window.

Find Next (Ctrl+N) finds the next instance of the text most recently searched.

Replace (Ctrl+H) permits us to find and replace text; this option is useful, for example, to change the names of the variables.

Go to line (Ctrl+T) to introduce the line number of the active window in which we want to place the cursor.

Match parenthesis (Ctrl+P) to find the closing parenthesis to that selected. It is useful in embedded sentences. If no parenthesis has been selected, LINGO chooses the one closest to the position of the cursor.

Paste function to insert functions at the current cursor position. Choose first the category and then the function from the menu.

Select Font... (Ctrl+J) to select the letter font of the active window or printer. Sometimes it is easier to display the document with courier font.

Insert New Object Lets you insert OLE objects in the active window, such as tables, equations, charts…

Links This option allows modifying the properties of the links to the external objects of a LINGO document.

Object Properties (ALT+Enter) If we select this option after selecting an external object, LINGO allows us to change the object's options.

LINGO Menu:

Solve (Ctrl+U) solves the model stored in the memory. If there is more than one, LINGO solves the model of the active window.

Solution (Ctrl+W) generates a solution report (text or graphical format) for the active window. We may want to see only the variables with nonzero values and/or only the constraints that are binding.

Range (Ctrl+R) generates the report of sensitivity analysis, giving the range of values in which it can:

1. Change one coefficient of the objective function without modifying the optimal values of the decision variables.

2. Change one coefficient of the RHS without modifying the optimal values of the opportunity costs and the reduced costs.

To enable range computations, select the *General Solver Tab* under *LINGO/Options* and in the Dual Computations list box, choose the *Prices & Ranges* option.

Options (Ctrl+I) allows us to change some parameters of the LINGO interface, as well as LINGO solve the model.

Interface shows a dialog box to control the appearance of LINGO, the output and the default file format.

General Solver allows us, among many other possibilities, to ckeck the box ***Variables assumed non-negatives*** to place a lower bound of zero on all variables. ***Dual computations*** **box** permits us to select ***Prices & Ranges*** to obtain sensitivity analysis report. In General Solver tab we can indicate time and number of iterations used to solve the model by ***Runtime Limits.***

Linear Solver provides options that allow configuring the way LINGO solves linear models. Among other possibilities we can choose primal simplex, dual simplex or interior point algorithm (Method/Primal simplex/Dual Simplex o Barrier). You can check a box to scale a model. The box Initial Linear Feasibility.Tol allows changing the tolerance value of the initial linear feasibility and by default it is 0.0000003. It is used at the initial stage of linear model solving in order to see whether a constraint is satisfied. Violations smaller than tolerance are ignored.

The box **Final Linear Feasibility.Tol** controls the tolerance value of the final solution feasibility and by default it is 0.00000001. It is used at the final stages of model solving to see if the constraints are satisfied. Violations smaller than tolerance are ignored.

Nonlinear Solver controls options that affect the algorithms to solve nonlinear models. Inicial Nonl Feasibility Tol. box indicates the tolerance of the initial nonlinear feasibility. By default, it is 0.001. Similarly, the box Final Nonl Feasibility.Tol provides the tolerance of the final nonlinear feasibility. By default, it is 0.000001. The two previous tolerances are used in a similar way to those used in linear models.

Integer Pre-Solver has options to reformulate the model in order to solve it as fast as possible with branch and bound algorithm. The integer pre-solver operates only with linear integer models.

Integer Solver tab provides options that allow us to configure the way in which LINGO solves integer programming models.

The Branching box has two options for controlling the branching strategy used in branch and bound algorithm. The Direction field controls how LINGO makes branching decisions (up, down or both).

Priority field permits us to decide if binary variables have priority in branching process.

Due to a round-off error on digital computers, it is not always possible to find integer values for integer variables. We can manage several options in Integrality box, such as ***Absolute Integrality*** and ***Relative Integrality***. The former tolerance is used as a test for integrality in integer programming models. This tolerance measures the difference between the variable value and an integer value. The latter tolerance is a similar concept measured in relative terms.

LP Solver box allows us to select the algorithm to use in branch and bound process (primal simplex, dual simplex and barrier solver).

Optimality box is used to control three tolerances. The Absolute Optimality tolerance is a positive value r, indicating to the branch and bound solver that it should only search for integer solutions with objective values at least r units better than the best integer solutions found so far.

The ***Relative Optimality tolerance*** is a similar concept to the previous tolerance. In this case r is ranging from 0 to 1, indicating that branch and bound should only search for integer solutions with the objectives values at least 100* r % better than the best integer solution found so far.

The ***Time to Relative*** tolerance is the number of seconds before the branch and bound solver begins using the Relative Optimality tolerance.

Finally the ***Tolerances*** box includes some tolerances for controlling the branching strategy: Hurdle, node selection and strong branch. Hurdle allows entering a known value of objective function. Then LINGO will only search for integer solution in which the objective is better than the hurdle value. The Node Selection option permits controlling the order in which the algorithm selects branch nodes in the tree (Depth First, Worst Bound and Best Bound).

Global Solver is an additional option of LINGO. It converts non-convex models into smaller convex models. It uses techniques such as linear programming and constraint propagation within a branch and bound framework to find global solution to non-convex models.

Generate generates the current model report, useful for model verification.

Picture represents the model in matrix form. This allows identifying repetitive structures in the model, and finding possible mistakes.

Debug is a command useful in the search for problems in both infeasible and unbounded linear models.

Model Statistics This option generates model statistics, such as number of variables, number of constraints, etc...

Look (Ctrl+L) generates a report containing the model formulation. You can see all or selected rows.

WINDOW Menu:

Command Window (Ctrl+1) opens an access window to the command line. In general, Windows users do not require this window.

Status Window (Ctrl+2) provides information about:

Total number of variables in the model, grouped as linear and nonlinear.

Status of the algorithm with the current status of the solution, the number of iterations performed, current sum of non-feasibilities, current value of the objective function, the best integer solution and the bound of the integer solution.

The number of constraints of the model, divided into linear and nonlinear.

The number of coefficients different from zero, divided into linear and nonlinear.

Current memory used to store data.

Time run to obtain the solution.

Furthermore, this window has an option to stop the solving process. In this case, it provides the best solution found up to that moment including the message that it may not be optimal or feasible. There is an option to close the window.

Finally, there is an option to indicate how often (in seconds) we want the window to be updated (**update interval**). In integer programming the window updates whenever a better integer solution is found, regardless of the value introduced. On the other hand, updating too often may increase solution times.

Send to back (Ctrl+B)

To send the active window to the back. For example, to move from the model window to the solution window.

Finally LINGO also has other common options in Windows, applications, such as Close All, Tile, Cascade and Arrange Icons.

Help menu allows us to access LINGO help.

A2.6. LINGO FUNCTIONS

LINGO has several types of functions as follows:

Standard operators, which are the arithmetic operators, the logical operators and the equality and inequality relationships.

Arithmetic operators: Power^, multiplication*, division /, addition+ and subtraction-.

Logical operators: #NOT#, #EQ#, #NE#, #GT#, #GE#, #LT#, #LE#, #AND# and #OR#.

Equality and inequality relations: =, ≤ and ≥. It also accepts < and > for less or equal and higher or equal, respectively. These relationships should not be confused with the logical operators #EQ#, #LE# and #GE#.

Variable domain functions: @BIN, @BND, @FREE, @GIN, @SEMIC, @SOS1, @SOS2, @SOS3, @CARD.

File Import: @IMPORT and @FILE. The latter is used when the data are stored in a file different from the model.

Financial functions: @FPA (I, N). @FPL (I,N).

Mathematical functions: @ABS, @COS, @EXP…

Set-Looping functions: @FOR, @MAX, @MIN, @SUM.

Probability functions: @PBN, @PCX…

ANNEX 3

MULTIPLE CRITERIA SOFTWARE FOR COLLABORATIVE DECISION MAKING: EXPERT CHOICE COMPARION SUITE

A3.1. MODELLING: DESIGN OF DECISION HIERARCHY

Expert Choice Comparion Suite is a web application designed for multiple criteria decision making both for a decision maker and for a working group. It uses the Analytic Hierarchy Process (AHP) and other methods to evaluate alternatives of a decision problem from the considered objectives (*Rating Scale, Utility Curves or Step Function*). Comparion facilitates tracking the preferences of all participants, their data and comments.

The participants or group members can be: Project owner, the only one that can create a project in the web application (it is enough to indicate a name), Project manager, who can create and modify the structure and options of the project or decision problem and the Project evaluator that may only issue the requested value judgments. The project director or the manager can collect qualitative and quantitative information from all evaluators.

Create a project

Only the Project owner will be able to create a project. To log in into Comparion CoreTM as project owner or project manager, use the e-mail address and the password and instructions received from Expert Choice. To introduce the data for a new problem, click on the *New Project* button on the main menu. Enter the name and the description of the project and click OK.

Decision hierarchy

Structuring the decision problem includes defining criteria and objectives, identifying alternatives, mapping alternatives to objectives and defining measurement methods.

Click on Structure in the main menu and Objectives on the secondary menu on the left of the screen. Select the goal of the project (decision problem). To enter the goal, right click on Goal and enter the goal description. Alternatively, you can click the left mouse button and enter the goal name. Additionally, you can introduce information about the project in the right-handwindow (Edit Information Document).

To add objectives, right click on goal and select "Add (level below)" in "Objectives" which is in the hand corner of the window. A window will pop up. Enter your objectives in the pop up window. Click OK to add the objectives. Add the description or other necesary information. You can also create sub-objectives by selecting the desired objective and clicking on Add (level below).

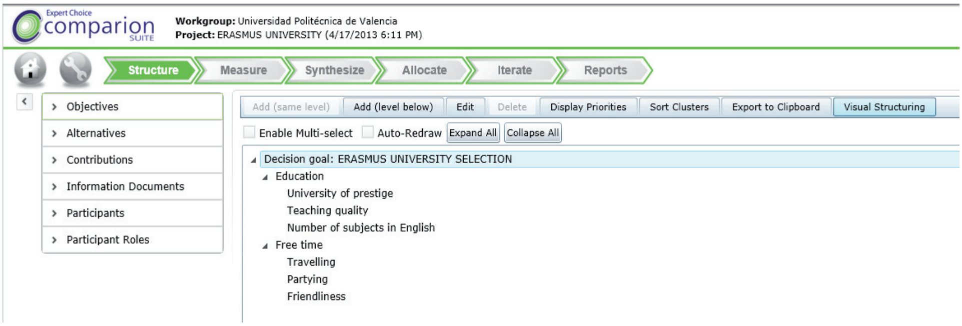


Figure A.3.1. Structure of the multiple criteria decision making problem

To add alternatives, right click on the word Alternatives on the main menu Structure and select Add to enter the alternatives one by one or select Paste from clipboard. Use this command to copy alternatives from another source (Figure A.3.1).

Alternative to objective mapping is useful if you have a structure where certain alternatives cannot be measured against some objectives. Full mapping is the default mapping, that is, all objectives are related to all alternatives. In other words, all of the alternatives contribute to reaching all of the objectives. To define the relation, click on Contributions on the left side of the window and uncheck (click the check mark) the box next to the alternatives that you do not want to evaluate against the highlighted objective.

A3.2. MULTIPLE CRITERIA METHODS

After introducing the objectives and the alternatives we have to define the evaluation multiple criteria methods. This can be done from the main menu, under Measure. The measurement method determines how each of the evaluation steps are presented to the evaluator or evaluators in the case of collaborative decision making.

The weight of the objectives can be obtained by pairwise comparison or direct assignment. Methods for evaluating the alternatives against the objectives of the lowest level of the hierarchy are as follows:

1. *Rating Scale*
2. *Pairwise Comparison*
3. *Utility Curves (Simple o Advanced)*
4. *Step Function*
5. *Direct Input*

Rating Scale presents a predefined qualitative scale from which evaluators select an option. By default, alternatives are evaluated using a rating scale measurement type. One example could be a scale which we consider excellent, very good, good and poor. Another example of a scale would be to consider the educational level of the staff: Doctorate, Master, and high school graduate. Keep in mind that once you have conducted evaluations of the alternatives in relation to a specific objective, changing the measurement types for that objective clears all of the judgments which have been made for those alternatives.

Pairwise Comparison provides evaluators with two objectives or alternatives that are compared to each other to determine their relative preference or importance with respect to the parent objective (Figure A.3.2). To select pairwise comparison for an objective, simply click on *Measurement Methods* and from the drop down menu, select *For Objectives*. You will be able to select the comparison method for each objective. The same procedure can be used in the case of the alternatives (*For Altenatives)*.

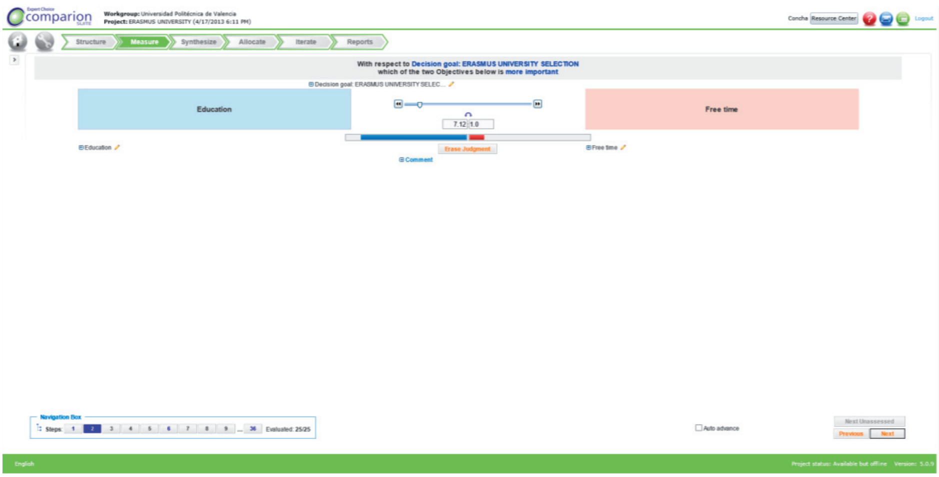


Figure A.3.2. Objectives Pairwise comparison

Utility Curve presents evaluators with a predefined curve of values to which they apply their data. You can use *Decreasing/Increasing* button to switch between decreasing and increasing utility curves.You can also change the values of the X-axis.
Once you have completed all of the comparisons, for example how to prioritize the Erasmus destination, Comparion gives you the option to review the judgements and shows the priorities both for the objectives and the alternatives, as we can see in figure A.3.3.

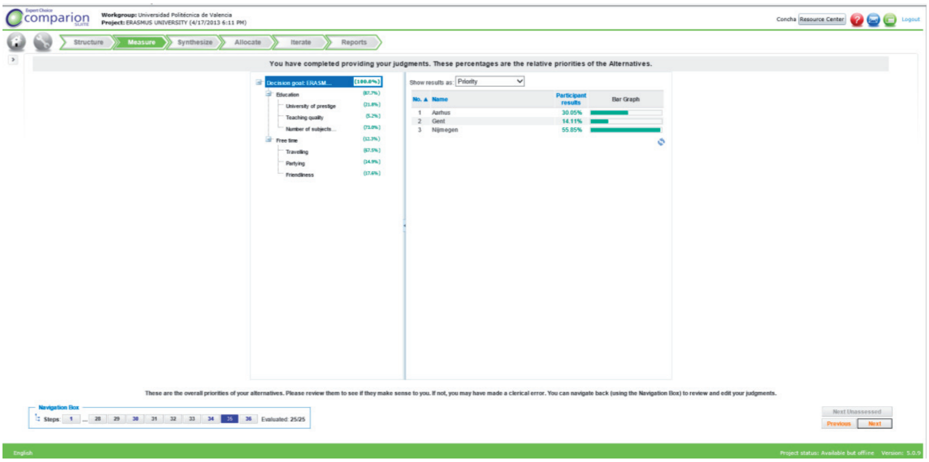


Figure A.3.3. Priorities for the objectives and Erasmus destinations for student 2

A3.3 COLLABORATIVE DECISION MAKING

Director, manager and evaluator

Once you have completed the decision hierarchy and have defined the evaluation method, the next step is to add and manage the participants. To define the evaluators go to Structure in the main menu, click on Participants in the secondary menu on the left hand side of the window and enter the e-mail address and name of the evaluator. Select the option Generate random password and then *Send a registration notification* to user to send the invitation to participate in the project.

Once you have done that, you can decide if this participant can manage the project (Project Manager) or only provide judgements. As Project Manager the person will be able to change criteria, alternatives, evaluation methods and invite other evaluators.

You can set the role of each evaluator. Setting a role for an evaluator means selecting specific objectives and alternatives for which only he/she can provide judgements. To set the evaluator role, first choose an evaluator from the dropdown menu. Click on the check box Participant Role and in For Objectives to select those objectives that should be evaluated by the evaluator. You can also select the alternatives to evaluate in the tab For Alternatives.

The director and the project manager can invite evaluators by clicking on the tab Anytime Evaluation / ***Invite Participant*** and provide their own assessment by clicking on the tab ***Collect my input***.

Collect assessments

There are several ways to send invitations to participants. The easiest way to send an e-mail to all evaluators is by using the Measure/Anytime evaluation/Invite Participants/Send Invite menu. If you need to edit the default text you can do so by clicking the Edit Invite button and writing in the body of the message that appears on the right side of the window. Use the Send Invite button to send invitations to all of the selected evaluators.

As a project owner (director or project manager) you can input your judgements by clicking on *Collect my input*. Press *Next* to proceed. You will be presented a series of prompts to collect your judgements. The evaluators simply need to log in to begin the evaluation (evaluators may need to select the project if they have been added as an evaluator to several projects).

A3.4 RESULTS ANALYSIS

View Results

To track the evaluation progress of your evaluators go to the main menu Synthesize, secondary menu Overall Results. Overall Results shows you the priorities of the objectives and the alternatives. The priorities of objectives shown here are calculated using the judgements entered by a single evaluator (clicking on ***Select users***). We can make groups or choose all of the evaluators, ***All participants*** (Figure A.3.4).

Sensitivity analysis

The sensitivity analysis is to analyze how the priorities of the alternatives vary when modifying the weights of the objectives. There are several types of graphs that help us to perform this analysis.

- Dynamic Analysis (***Dynamic Sensitivity***): The priorities of the objectives and alternatives are represented by a bar graph. Lengthening or shortening the bars of the objectives alters the alternatives bars (Figure A.3.5.)

- ***Performance SensitivityAnalisys***: This analysis is a 3-axis graph. On the X-axis are the objective names. On the Y-axis is a 0-100% scale to indicate the priorities of the objectives. The bars for each objective are partially filled with light blue to indicate the priority of each objective. The priority for the respective objective is also printed below each bar. On the Z-axis is a scale that indicates the priorities of alternatives. You can drag the bars to change the priorities of the objectives and see how the priorities of the alternatives change.

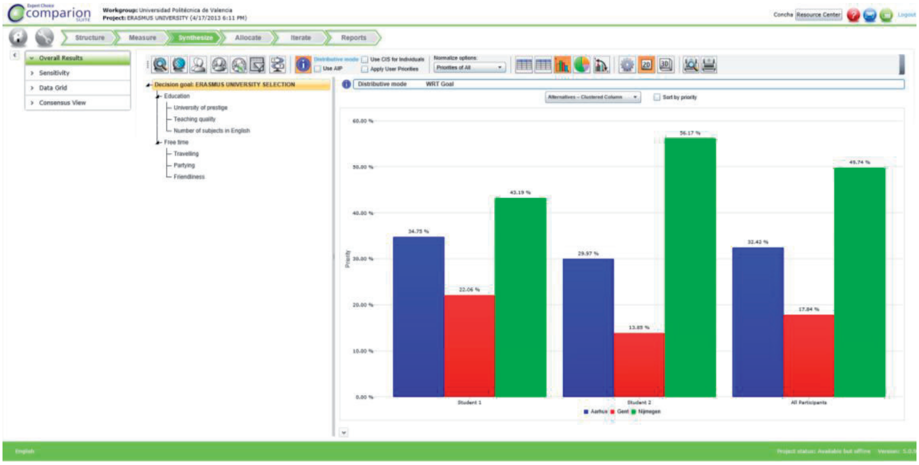


Figure A.3.4 Individual and aggregated results of students

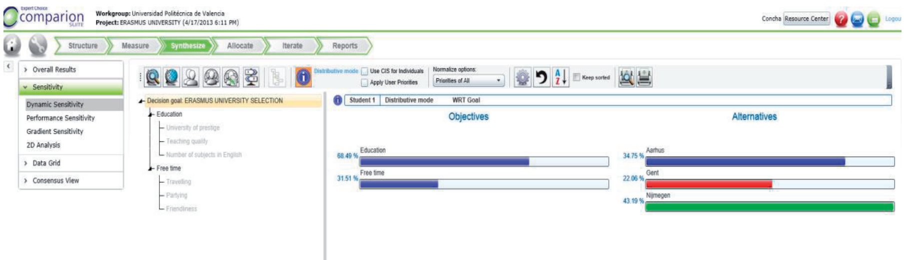


Figure A.3.5. Dynamic sensitivity analysis

- ***Gradient sensitivity Analysis***: You can change the priority of a specific objective and see how the priorities of the alternatives change. Select the objective in the drop down menu. The points of intersection of other lines with the blue line give the priority of alternatives for the specific priority of the selected objective. You can drag the blue line to perform the sensitivity analysis. The red line represents the base priority of the objective selected in the drop down menu (Figure A.3.6.)

- **Graph 2D**: represents the alternatives in a two-dimensional graph in which the objectives can be chosen, weights of which are shown in the x and y axes.

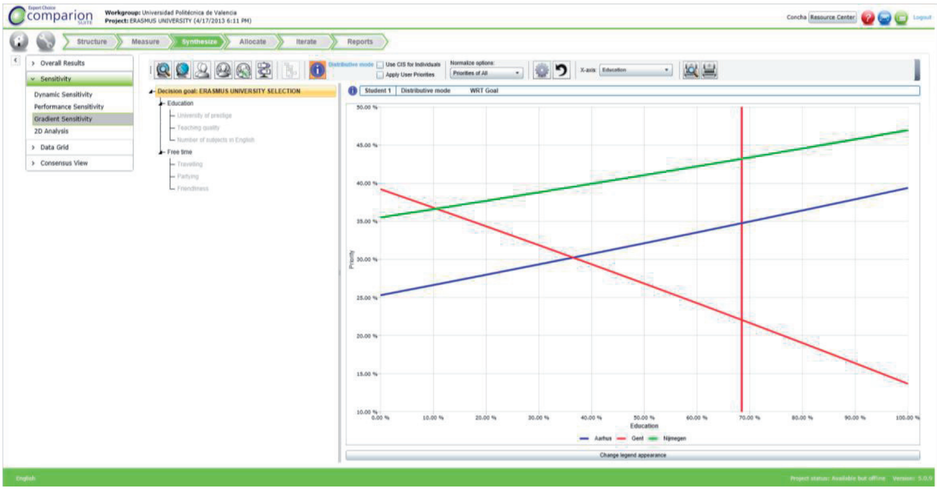


Figure A.3.6. Gradient Sensitivity Analysis

Generate Reports

Comparion Core provides an exhaustive report generating system to reports for every judgement entered by each participant (project owner, project manager and evaluator). Select *Reports* in the main menu to see the links *Ad-Hoc Reports* and *Predefined Reports*. *Predefined Reports* include the following reports: *Objectives and Alternatives Contributions of Alternatives to Objectives, Overview of inputs, Priority of Objectives, Priority of Alternatives, Objective/Alternative Priorities,* and *Overall Results*. You can print them and save them in several formats.

The reports under *Ad-Hoc Reports* are customizable. For example, if you click on *Objectives Priorities*, you can click *User* to select/de-select users whose priorities you want to see. Similarly, you can choose any objective or alternative whose report you want to view. Click on *Judgement Overview* to see the judgements entered by all participants. You can customize this report by selecting a particular set of participants, objectives and alternatives to include in reports.

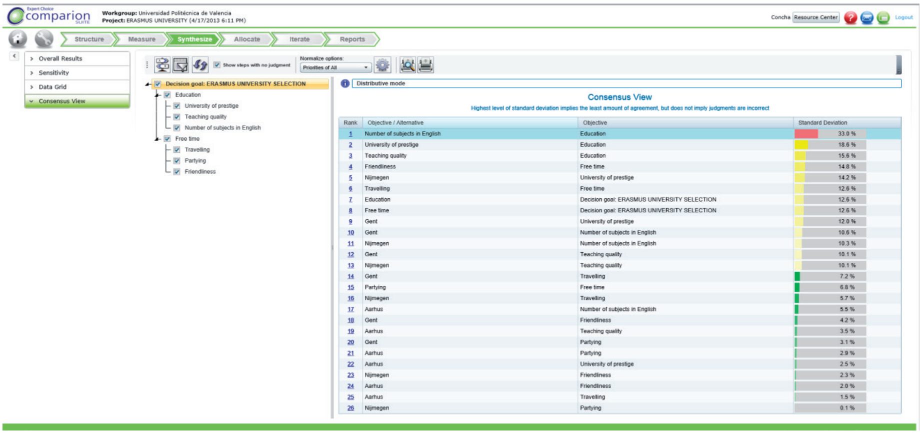


Figure A.3.7 Consensus Analysis