Energy efficiency, robustness, and makespan optimality in job-shop scheduling problems

Article *in* Artificial intelligence for engineering design analysis and manufacturing · June 2015

DOI: 10.1017/S0890060415000335

Miguel A. Salido
Universitat Politècnica de València
147 PUBLICATIONS **1,299** CITATIONS

Joan Escamilla
Universitat Politècnica de València
13 PUBLICATIONS **71** CITATIONS

Federico Barber
Universitat Politècnica de València
152 PUBLICATIONS **1,195** CITATIONS

A. Giret
Universitat Politècnica de València
147 PUBLICATIONS **1,264** CITATIONS

Energy-efficiency, Robustness and Makespan Optimality in Job-shop Scheduling Problems

Miguel A. Salido[1], Joan Escamilla[1], Federico Barber[1], Adriana Giret[1],
Dunbing Tang[2], Min Dai[2]
[1]Instituto de Automática e Informática Industrial
Universitat Politecnica de Valencia, Spain.
[2]College of Mechanical and Electrical Engineering
Nanjing University of Aeronautics and Astronautics, China.

January 13, 2015

Abstract

Many real-world problems are known as planning and scheduling problems, where resources must be allocated so as to optimize overall performance objectives. The traditional scheduling models consider performance indicators such as processing time, cost and quality as optimization objectives. However, most of them do not take into account energy consumption and robustness. We focus our attention in a job-shop scheduling problem where machines can work at different speeds. It represents an extension of the classical job-shop scheduling problem, where each operation has to be executed by one machine and this machine can work at different speeds. The main goal of the paper is focused on the analysis of three important objectives: Energy-efficiency, Robustness and Makespan, and the relationship among them. We present some analytical formulas to estimate the ratio/relationship between these parameters. It can be observed that there exist a clear relationship between robustness and energy-efficiency and a clear trade-off between robustness/energy-efficiency and makespan. It represents an advance in the state of the art of production scheduling, so obtaining energy-efficient solutions also supposes obtaining robust solutions, and viceversa.

1 Introduction

Nowadays, the main objective of many companies and organizations is to improve profitability and competitiveness. These improvements can be obtained with a good optimization of resources allocation. But in the last years many companies are not only facing complex and diverse economic trends of shorter product life cycles, quick changing science and technology, increasing customer demand diversity, and production activities globalization, but also enormous and heavy environmental challenges of global climate change (e.g. greenhouse effect), rapid exhaustion of various non-renewable resources (e.g. gas, oil, coal), and decreasing biodiversity.

Scheduling problems are widely discussed in the literature and two main approaches can be distinguished [2]:

- Classical deterministic methods, which consider that the data are deterministic and that the machine environment is relatively simple. Some traditional constraints are taken into account (precedence constraints, release dates, due dates, preemption, etc.). The criterion to optimize is often standard (makespan). A number of methods have been proposed (exact methods, greedy algorithms, approximate methods, etc.), depending on the difficulty of a particular problem. These kinds of studies are the most common in the literature devoted to scheduling problems.

- On-line methods. When the algorithm does not have access to all the data from the outset, the data become available step by step, or ”on-line”. Different models may be considered here. In some studies, the tasks that we have to schedule are listed, and appear one by one. The aim is to assign them to a resource and to specify a start time for them. In other studies, the duration of the tasks is not known in advance.

Flexibility occurs at the boundary between these two approaches: some information is available concerning the nature of the problem to be solved and concerning the data. Although this information is imperfect and not wholly reliable, it cannot be totally ignored. It is well-known that there will be discrepancies, for a number of reasons, between the initial plan and what is actually realized. Given a set of disruptions that can occur in unforeseen circumstances, the aim is to propose one or more solutions that adapt well to disruptions, and then produce reactive decisions in order to ensure a smooth implementation [2].

The job-shop scheduling problem (JSP) represents a problem where there are some specific resources or machines which have to be used to carry out some tasks. Many real life problems can be modeled as a job-shop scheduling problem and can be applied in some variety of areas, such as production scheduling in the industry, departure and arrival times of logistic problems, the delivery times of orders in a company, etc.

Most of the solving techniques try to find the optimality of the problem for minimizing the makespan, minimizing tardiness, minimizing flow-time, etc.

Recently some works have focused on minimizing the energy consumption in scheduling problems [19][8], mainly from the Operations Research Community [5], [20]. Furthermore some works have been carried out to obtain robust schedules to absorb incidences in dynamic scheduling [6]. However, no works have related energy-efficiency with robustness although there exist a clear relationship between them in job-shop scheduling.

We focus our attention in a job-shop scheduling problem with different speed machine (JSSM). It represents an extension of the classical job-shop scheduling problem, where each operation must be executed on a machine at a determined speed (by a classical deterministic method). If the speed of a machine is high, the energy consumption increases, but the processing time of the task decreases, meanwhile if the speed is low, the energy consumption decreases and the processing time increases. Thus, in on-line scheduling, if disruptions occur, reactive decisions are needed, so machines can accelerate the speed to absorb these disruptions and recover the original schedule, obtained by the classical deterministic method.

To this end, we analyze the relationship between some important parameters in order to obtain a multi-objective solution. We show that there is a clear tradeoff between Makespan and Energy consumption; and between Makespan and Robustness. Therefore there is a close relationship between Energy consumption and Robustness. However, this close relationship has not being analyzed in the literature and new techniques can be developed to achieve these objectives jointly. Thus, our main goal is to find a solution that minimizes the energy consumption and the makespan. Furthermore, we extend this goal to determine the saved time by energy efficiency as a robustness measure in order to be used if incidences appear. Thus, if a task is delayed, the lost time can be recovered by increasing the speed of the machine to recover the original solution.

2 Problem Description

Formally the job-shop scheduling problem with different speed machine (JSSM) can be defined as follows. We are given a set of n jobs $\{J_1, \dots, J_n\}$, a set of m resources or machines $\{R_1, \dots, R_m\}$. Each job J_i consists of a sequence of v_i tasks $(\theta_{i1}, \dots, \theta_{iv_i})$. Each task θ_{il} has a single machine requirement $R_{\theta_{il}}$ and a start time $st_{\theta_{il}}$ to be determined. Each machine can work with different speeds, so each task is linked up to an integer duration $p_{\theta_{il}}$ and an integer energy $e_{\theta_{il}}$ used by the corresponding machine.

A feasible schedule is a complete assignment of starting times to tasks that satisfies the following constraints: (i) the tasks of each job are sequentially scheduled, (ii) each machine can process at most one task at any time, (iii) no preemption is allowed. The objective is finding a feasible schedule that minimizes the completion time of all the tasks and the energy used. The problem is a standard job-shop problem denoted as $J||C_{max}$ according to classification scheme proposed in [3]. But the association between duration and energy have been created so the problem JSSM can be denoted as J(Speed)//Cmax,Energy. For each task, three different speeds have been defined. Each speed has a duration and an energy consumption. When the working speed increases the energy also increases but the duration decreases.

3 Energy Efficiency

Nowadays manufacturing enterprisers are not only facing complex and diverse economic trends of shorter product life cycles, quick changing science and technology, increasing customer demand diversity, and production activities globalization, but also enormous and heavy environmental challenges of global climate change (e.g. greenhouse effect), rapid exhaustion of various non-renewable resources (e.g. gas, oil, coal), and decreasing biodiversity. Statistical data in 2009 shows the Germany industrial sector was responsible for approximately 47% of the total national electricity consumption, and the corresponding amount of CO2 emissions generated by this electricity summed up to 18%-20% [4]. Thus, manufacturing companies are responsible for the environmental outcome and are forced to have manufacturing systems that demonstrate major potential to reduce environmental impacts [9].

Recently, there has been growing interest in the development of energy savings due to a sequence of serious environmental impacts and rising energy costs. Research on minimizing the energy consumption of manufacturing systems has focused on three perspectives: the machine level, the product level, and the manufacturing system level. From the machine-level perspective, developing and designing more energy-efficient machines and equipment to reduce the power and energy demands of machine components is an important strategic target for manufacturing companies [18][21]. Unfortunately, previous studies show that the share of energy demand for removal of metal material compared to the share of energy needed to support various functions of manufacturing systems is quite small (less than 30% of total energy consumption) [7][12].

From the product-level perspective, modeling embodied product energy framework based on a product design viewpoint for energy reduction approach is beneficial to support the improvements of product design and operational decisions [23][26]. It requires strong commercial simulation software to facilitate the analysis and evaluation of the embodied product energy. The results cannot be applied easily in most manufacturing companies, especially in small- and medium-size enterprises due to the enormous financial investments required. From the manufacturing system-level perspective, thanks to decision models that support energy savings, it is feasible to achieve a significant reduction in energy consumption in manufacturing applications. In the specialized literature about production scheduling, the key production objectives for production decision models, such as cost, time and quality have been widely discussed. However, decreasing energy consumption in manufacturing systems through production scheduling has been rather limited. One of the most well-known research works is the work of Mouzon et al.[20], who developed several algorithms and a multiple-objective mathematical programming model to investigate the problem of scheduling jobs on a single CNC machine in order to reduce energy consumption and total completion time. They pointed out that there was a significant amount of energy savings when non-bottleneck machines were turned off until needed; the relevant share of savings in total energy consumption could add up to 80%. They also reported that the inter-arrivals would be forecasted and therefore more energy-efficient dispatching rules could be adopted for scheduling.

In further research, Mouzon and Yildirim [19] proposed a greedy randomized adaptive search algorithm to solve a multi-objective optimization schedule that minimized the total energy consumption and the total tardiness on a machine. Fang et al. [10] provided a new mixed-integer linear programming model for scheduling a classical flow shop that combined the peak total power consumption and the associated carbon footprint with the makespan. Bruzzone et al. [5] presented an energy-aware scheduling algorithm based on a mixed-integer programming formulation to realize energy savings for a given flexible flow shop that was required to keep fixed original job assignment and sequencing.

Although the majority of the research on production scheduling has not considered energy-saving strategies completely, the efforts mentioned above provide a starting point for exploring an energy-aware schedule optimization from the viewpoint of energy consumption. However no work has been carried out to consider a multi-objective optimization schedule to minimize the total energy consumption, the makespan and to maximize the robustness of the schedule.

4 Robustness

Robustness is a common feature in real life problems. Biological life, functional systems, physical objects, etc, persist if they remain running and maintain their main features despite continuous perturbations, changes, incidences or aggressions [24]. Thus, robustness is a concept related to the *persistence* of the system, of its structure, of its functionality, etc., against external interferences: *A system is robust, if it persists*.

It is really difficult to give a unique definition for robustness, as this concept is differently defined in several domains. Furthermore, the definition often remains implicit in the literature or is determined by the specific target application. Finally, most authors prefer to use the concept of robust solution (and here, of robust schedule).

The data associated with a scheduling problem are the processing times, occurrence dates of some events, some structural features and the costs. None of this data is free from factors of uncertainty. The duration of tasks depends on the conditions of their execution, in particular on the necessary human and material resources. They are thus inherently uncertain, regardless of contingent factors that may impair their execution. For instance, transportation times for components between separate operations in a manufacturing system will depend on the characteristics of the transportation resources available. Finally, in a production scheduling, some resources such as versatile machines require a reconfiguration time between operations. This time depends on the type of tools needed and the location of these tools in the shop, not to mention the operator carrying out the reconfiguration [2].

Let us first propose some consensus definition: a schedule is robust if its performance is rather insensitive to the data uncertainties. Performance must be understood here in the broad sense of solution quality for the person in charge; this naturally encompasses this solution value relatively to a given criterion, but also the structure itself of the proposed solution. The robustness of a schedule is a way to characterize its performance.

In the literature, it is sometimes difficult to separate sensitivity analysis and robustness. In fact the sensitivity analysis tries to answer the what if... questions. It deals with disturbances more than with general uncertainty: data are fixed but might be disturbed [2].

In scheduling problems, Robustness can be defined as:

Definition 1: ***Robustness***: is the ability of a solution to maintain its feasibility when incidences appear during execution in the scheduling problem.

In this paper the robustness of a schedule will be used to answer what if... questions, mainly related to small disruptions that daily occur in real life scheduling problems. In this way, the robustness of a schedule can be used to obtain energy-aware schedules that do not modify the start time of tasks. To this end, the slack between tasks that makes the schedule robust (to absorb incidences) can be profitable by machines to work at lower speed and therefore saving energy consumption. However if this slack is needed, due to a disruption, the involved machine can increase its speed in order to recover the disrupted time and finalize the task on time. In this way, there exist a relationship between robustness and energy saving that can be applied to many scheduling contexts.

5 Modeling and Solving a JSSM

The more natural way to solve the Jop-Shop Scheduling Problem involves all variables and constraints related to jobs, tasks and machines [11] [13] [22]. However, the solution obtained is an optimal solution that minimizes the makespan but it does not guarantee a certain level of robustness. Generally, this solution is not able to absorb incidences and a delay in a task is propagated along the rest of the schedule.

Several reactive/proactive techniques have been developed in the literature to manage incidences in scheduling problems [2]. Thus, computing a new solution from scratch after each problem change is possible (reactive technique), but it has two important drawbacks: inefficiency and instability of the successive solutions [25]. Whilst reactive methods merely deal with the consequences of an unexpected change, taking a more proactive approach may guarantee a certain level of robustness. We are interested in this proactive approach so that our goal is searching for a equitable trade-off between robustness and optimality of a solution.

Robustness (as in section 4) in job-shop scheduling can be obtained through allocating buffer times between tasks in order to absorb small disruptions (task delays, etc.) that can occur stochastically along the schedule.

In an optimized solution of a JSSM, some natural buffers appear to satisfy the involved constraints (non-overlapping constraints). These buffers give the schedule some robustness degree. However, if more buffers must be included to make the final solution more robust, the involved tasks must be moved and the effect must be propagated to the rest of the schedule.

To add robustness to JSSM solutions, we use the extra speed that machines can work only in cases where machines are not working at top speed. Several solutions are obtained with different weights to minimize makespan or energy used. When the main objective is to minimize the energy used, the solutions are composed of several tasks that are processed by machines in a low speed. If some incidences appear this speed can be increased and the solutions remain valid. Following this idea the energy roominess can be considered as robustness.

5.1 IBM ILOG CPLEX CP Optimizer tool

The problem is modeled and solved with IBM ILOG CPLEX CP Optimizer tool (CP Optimizer). CP Optimizer uses constraint programming technology to solve detailed scheduling problems and other hard combinatorial optimization problems [15].

CP Optimizer is a commercial solver embedding powerful constraint propagation techniques and a self-adapting large neighborhood search method dedicated to scheduling [17]. This solver is expected to be very efficient for a variety of scheduling problems as it is pointed in [14], in particular when the cumulative demand for resources exceeds their availability as it happens, for example, in the Satellite Control Network Scheduling Problem confronted in [16].

The problem has been modeled as a typical job-shop scheduling problem. The extension with different machine speeds has been implemented considering that each task is executed by a machine and this machine has different optional modes where each mode represents the duration of the task and an associated consumption energy.

The objective is to find a solution that minimize the multi-objective makespan and energy consumption. The weight of each objective can be changed by λ parameter. Following the expression (1).

$$\lambda * Makespan + (1 - \lambda) * EnergyConsumption \quad (1)$$

Since the values of energy consumption and makespan are not proportional, it is necessary to normalize both measures (NormEnergy)(NormMakespan). NormEnergy value is calculated by summing the energy used in the execution of all the tasks, divided the maximum energy (maxEnergy). maxEnergy is the sum of the energy needed to execute all task at top speed. The NormMakespan is the makespan divided the sum of the task durations when the machines are working at the lowest speed (maxMakespan). The objective function is the expression (2).

$$\lambda * NormMakespan + (1 - \lambda) * NormEnergy \quad (2)$$

Algorithm 1: Model in CP-Optimizer

```
Data: $tasks$: Set of tasks; $modes$: Set of 3 modes for each task;
Result: A solution minimizing the objective function depending of $\lambda$
$Interval\ itvs$ := Define interval, one for each task;
$Interval\ modes$ := Define mode, three for each task;
$Sequence\ mchs$ := Each itvs with the same machine is linked up;
Minimize($(1 - \lambda) * NormEnergy + \lambda * NormMakespan$)
Subject to
    • noOverlap ($mchs$)
    • endBeforeStart ($itvs[j][o]$, $itvs[j][o+1]$)
    • alternative ($itvs[j][o]$, all(md in $modes$: if(md.id == itvs.id)))
```

Algorithm 1 shows a pseudo-code of the model to solve the problem.

Figure 1 shows two different schedules obtained by CP Optimizer for a given instance of the JSSM proposed in [1]. This instance represents a scheduling problem with 3 machines, 3 jobs, each with 5 tasks and each task has a processing time between 1 and 10 time units when the machine works at full speed. Each task is represented by a grey rectangle which can be divided in two regions: a solid black grey color represents the processing time when the machine is working at full speed (mandatory), and a light grey color with horizontal lines represents the extra processing time if the machine doesn't work at full speed (optional). This region represents the used time to save energy. However this time can also be used to absorb incidences if a disruption occurs

(EEBuffer). Each task is labeled with the number of task, the machine used and the speed used by the corresponding machine (green:low speed, yellow: medium speed, red:full speed). Finally the black rectangles represent natural buffer times generated by the solution. They can also be used to absorb incidences.

Two solutions (schedules) have been obtained for the same instance with different lambda values (λ) between 0.1 and 0.9.

For $\lambda = 0.1$, the solution X_1 give more importance (0.9) to energy efficiency and less importance (0.1) to minimize makespan. It can be observed that the makespan was 54, no tasks was carried out by machines at full speed (red), 2 tasks at medium speed (yellow) and 13 tasks at low speed (green). It generated an energy consumption of 79 units and it can be observed that all tasks maintain slack to absorb incidences so the robustness of the schedule is considered high.

For $\lambda = 0.9$, the solution X_2 gives more importance (0.9) to minimize makespan and less importance (0.1) to energy efficiency. It can be observed that the makespan was 34, and 7 tasks were carried out by machines at full speed (red), 4 tasks at medium speed (yellow) and 4 tasks at low speed (green). It generated an energy consumption of 112 units and it can be observed that only 8 tasks maintain slack to absorb incidences so the robustness of the schedule is considered low.

By modifying the value of λ, an approximate Pareto front for the bicriteria optimization schedule is generated. It must be taken into account that no single solution between the X_1 and X_2 can be said, a priori, to be the best one. Indeed, they are non-comparable, so choosing a solution from an approximate Pareto front can only be done by the user, depending on the requirements. This is why we advocate producing, for a given problem instance, the Pareto front rather than a single solution.

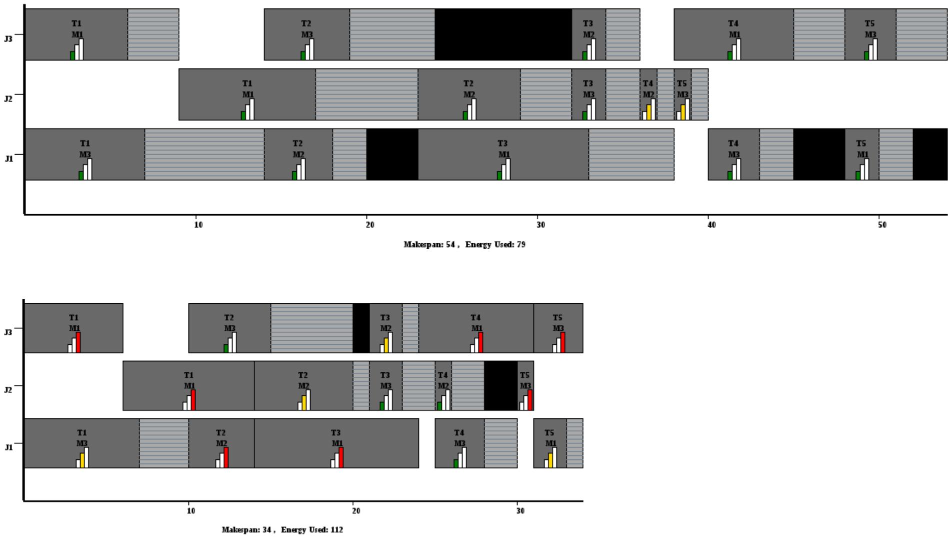

Figure 1: Schedules with different λ values

6 Definition of the Benchmark Set

To analyze the relationship among makespan, Energy consumption and robustness, we have evaluated the behavior on the benchmarks proposed in [1]. According on the benchmark results (small and large instances) several analytical formulas have been developed to estimate these parameters. All analyzed instances are characterized by the number of machines (m), the maximum number of tasks by job (v_{max}) and the range of processing times (p). The number of jobs (j) is set to 3. A set of instances was generated by combining values of each parameter: $m = 3, 5, 7$; $v_{max} = 5, 7, 10, 20, 25, 30$ and $p = [1, 10], [1, 50], [1, 100], [1, 200]$. In these benchmarks, the number of operators was not considered so that we fixed it to the number of machines. We have modeled the instances to be solved by the Optimizer.

We have also extended the original instances of Agnetis [1] to add different energy consumptions (e_1, e_2, e_3) to each task according to three processing times (pt_1, pt_2, pt_3), where pt_1 is equal to the original value of processing time in the Agnetis instances. pt_2 and pt_3 were calculated following the expressions (3) and (4), respectively. These instances can be found in our webpage[1].

$$pt_2{=}Max(maxdur{*}0.1{+}pt_1{,}Rand(1.25{*}pt_1{,}2.25{*}pt_1)) \tag{3}$$

$$pt_3{=}Max(maxdur{*}0.1{+}pt_2{,}Rand(1.25{*}pt_2{,}2.25{*}pt_2)) \tag{4}$$

The value $maxdur$ represents the maximum duration of a task for the corresponding instance and the expression $rand$ represents a random value between both expressions. Similar expressions were developed to calculate the energy consumption (5, 6, 7).

$$e_1{=}Rand(pt_1{,}3{*}pt_1)) \tag{5}$$

$$e_2{=}Max(1{,}Min(e_1{-}maxdur{*}0.1{+}{,}Rand(0.25{*}e_1{,}0.75{*}e_1)) \tag{6}$$

$$e_3{=}Max(1{,}Min(e_2{-}maxdur{*}0.1{+}{,}Rand(0.25{*}e_2{,}0.75{*}e_2)) \tag{7}$$

Following these expressions the processing times of pt_1, pt_2, pt_3 increase as the energy consumption of e_1, e_2, e_3 decrease. For example, given an instance with 5 tasks per job, three triplets are represented for each task: the id of the task, the energy used and the processing time ($< id, e, pt >$):

$< id, e_3, pt_3 >, < id, e_2, pt_2 >, < id, e_1, pt_1 >$
$< 1, 14, 14 >, < 1, 16, 10 >, < 1, 19, 7 >,$
...
$< 15, 3, 6 >, < 15, 5, 4 >, < 15, 6, 3 >,$

7 Makespan versus Energy Consumption

In this section we analyze the tradeoff between makespan and Energy consumption in job-shop scheduling problems with different machine speeds. Figure 2 shows an

[1] http://gps.webs.upv.es/jobshop/

approximate Pareto Front for a set of 10 instances with 5 machines, 10 tasks per job and a maximum processing time of 50 time units. For $\lambda = 1$, it can be observed that the average energy consumption was 1311 and the average makespan was minimized (317). However for $\lambda = 0$, the average energy consumption was minimized (745) and the average makespan was maximized (564.4). As we pointed out above, depending on the user requirements, a value of λ must be selected to obtain the desired level of makespan/energy consumption. Table 1 shows the makespan and energy consumption for each value of λ in different instances. It must be taken into account the relationship/ratio between makespan and energy consumption is similar in all instances, so that this tradeoff is not dependent on the number of machines, number of tasks per job, neither the range of processing times.

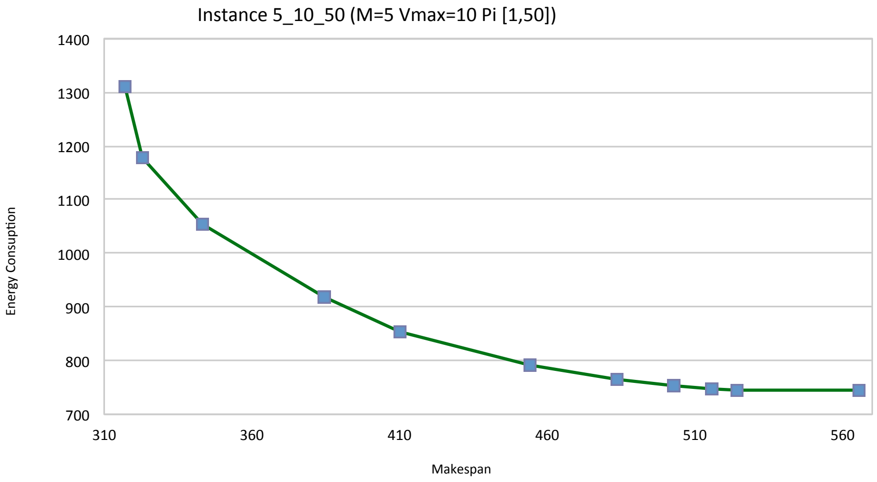

Figure 2: Approximate Pareto Front for the bicriteria Makespan-Energy Consumption.

Table 1: Makespan and Energy Consumption in instances $< m, v_{max}, p >$

	5_10_50		**7_10_100**		**3_20_50**		**3_25_100**		**3_30_200**	
λ	**Mk**	**Energ.**	**Mk**	**Energ.**	**Mk**	**Energ.**	**Mk**	**Energ.**	**Mk**	**Energ.**
0	565.4	745	1088.4	1571.4	1296	1507.4	3160	3827.1	7289.6	9162.7
0,1	524.4	745	1004.3	1571.6	1168.7	1507.4	2768.4	3827.5	6600.8	9163.5
0,2	515.8	747.1	992.1	1574.5	1145.7	1513.7	2734.2	3835.7	6513.2	9184
0,3	502.6	752.3	970.2	1584.2	1112.6	1527.3	2667.1	3866.7	6364.8	9258.8
0,4	483.6	764.2	918.3	1616.3	1079	1559.9	2553	3946.1	6158	9386.5
0,5	454.3	792.1	884.5	1650.3	1011.7	1628.6	2421.8	4077	5825	9771.5
0,6	410.1	854.1	843.7	1709.8	946.8	1722.5	2280.6	4300.6	5452	10239.4
0,7	384.7	917.4	768.2	1879.5	854.6	1933.7	2056.1	4742.9	4953.3	11370
0,8	343.1	1053	690.1	2147.3	777.8	2216.3	1854.4	5466.1	4530.8	12935.4
0,9	322.8	1179.1	635.7	2466.4	728.3	2498	1738.4	6164.9	4182.9	14696.1
1	317	1311.5	625.9	2664.1	707.8	2764.3	1686.1	6667.1	4048.9	16235.5

According to the analyzed instances, the ratio between energy consumption and makespan can be estimated by using the formula (8):

$$\frac{Energy(\lambda)}{Makespan(\lambda)} \approx e^{0,25+1,2\lambda} - \frac{sin(\lambda\pi)}{2} + \frac{cos(\lambda\pi)}{8} \qquad (8)$$

Thus, given a schedule instance with a given makespan and a λ value, we can estimate the energy consumption required to execute this schedule. In the same way, given a schedule instance with a given energy consumption threshold and a λ value, we can estimate the makespan needed o execute this schedule. This formula can be redefined by the operator according on the distribution of energy consumption of machines at different speeds. This formula and further formulas have been empirically obtained by approximation of all analyzed benchmarks. Firstly, they were approximated by polynomial interpolation and then, they were empirically approximated to a more complex formula to adjust the behavior in all desired points. Thus, they show that there is a clear relationship between the involved parameters.

8 Robustness versus Energy Consumption

The main goal of this paper is to show the tradeoff between robustness and energy consumption. In this way, the advantage could be twofold. By developing new techniques for searching energy efficient schedules also mean searching for robust schedules. Thus, these techniques will generate energy-aware and robust solutions in production scheduling, so small disruptions can be repaired by accelerating the needed machines to recover the original schedule. In this way, no rescheduling is needed and the user can adjust the parameters to obtained the optimal solution based on the problem preference.

To carry out this study, we have simulated 100 incidences to each instance in order to analyze the amount of incidences that can be absorbed by the resultant schedule. An incidence is a delay to a random task of the schedule. The duration of the incidence (%incid) was bounded by a 20% of total duration of the involved task. Figure 3 shows an approximate Pareto Front for a set of 10 instances with 7 machines, 10 tasks per job and a maximum processing time of 100 time units. It can be observed that as the robustness increased the energy consumption decreased. This is due to the fact that more robust solutions allow machines to work at minimum speed, so the energy consumption decreased, i.e. if all machines work at minimum speed, all tasks have a slack (time between solving the task at minimum speed minus solving the task at minimum speed). Thus, if a disruption occurs in a machine m_i at speed (s_{i1}) during the task t_i, this machine can accelerate its speed to s_{i2} in this task t_i in order to finish on time (before the next task t_{i+1} starts). In this case, we consider the schedule is robust. If the delay of task t_i affects to the following task t_{i+1}, the machine m_j that works in this task accelerates its speed in order to finish on-time. Finally, the disruption is absorbed in some steps. In this case, we consider the schedule is stable due to the fact that the disruption has been propagated to some other tasks before the original solution is recovered.

Table 2 shows the energy consumption and robustness in different instances. It must be taken into account that the robustness maintained the same behaviour in all instances, so that the robustness is not directly dependent on the number of machines,

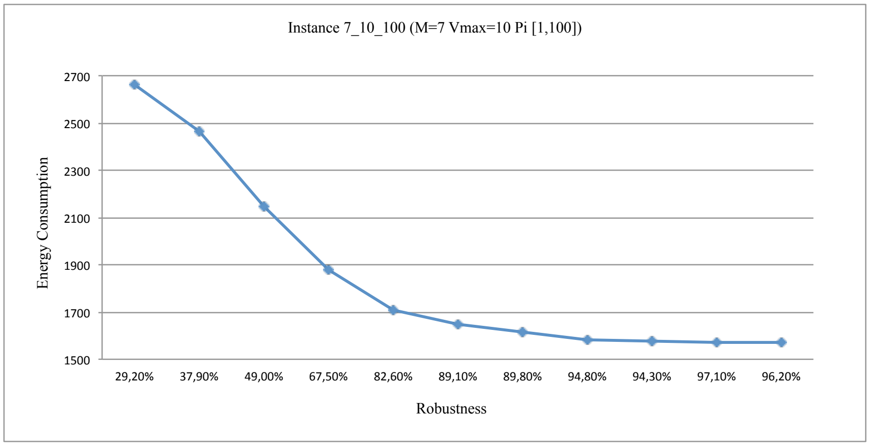

Figure 3: Approximate Pareto Front for the bicriteria Energy Consumption-Robustness.

Table 2: Energy Consumption and Robustness in instances $< m, v_{max}, p >$

	5_10_50		**7_10_100**		**3_20_50**		**3_25_100**		**3_30_200**	
λ	**Ener.**	**Robust.**	**Ener.**	**Robust.**	**Ener.**	**Robust.**	**Ener.**	**Robust.**	**Ener.**	**Robust.**
1	1311.5	26.7%	2664.1	29.2%	2764.3	26.8%	6667.1	27.0%	16235.5	25.6%
0,9	1179.1	37.2%	2466.4	37.9%	2498	36.3%	6164.9	36.8%	14696.1	39.7%
0,8	1053	48.2%	2147.3	49.0%	2216.3	50.6%	5466.1	49.5%	12935.4	48.9%
0,7	917.4	68.1%	1879.5	67.5%	1933.7	66.0%	4742.9	67.9%	11370	65.2%
0,6	854.1	79.8%	1709.8	82.6%	1722.5	79.6%	4300.6	78.9%	10239.4	80.9%
0,5	792.1	86.6%	1650.3	89.1%	1628.6	88.0%	4077	86.3%	9771.5	88.0%
0,4	764.2	91.8%	1616.3	89.8%	1559.9	93.4%	3946.1	94.1%	9386.5	92.7%
0,3	752.3	94.2%	1584.2	94.8%	1527.3	95.3%	3866.7	94.9%	9258.8	95.9%
0,2	747.1	97.0%	1574.5	94.3%	1513.7	96.6%	3835.7	96.0%	9184	96.8%
0,1	745	93.7%	1571.6	97.1%	1507.4	97.3%	3827.5	97.6%	9163.5	96.3%
0	745	97.2%	1571.4	96.2%	1507.4	99.0%	3827.1	98.9%	9162.7	99.2%

number of tasks per job, neither the range of processing times. However in most instances, for $\lambda = 0,1$ and $\lambda = 0$ the energy needed is similar, but the robustness is different (see last two rows in instances $3-5-10$, $3-7-10$ and $5-10-50$). Thus, given an energy consumption threshold, we can obtain different solutions with different robustness and makespan level.

The relationship between energy consumption and robustness can be estimated by using the formula (9):

$$\frac{Energy(\lambda)}{Robustness(\lambda, \%incid)} \approx \frac{ttasks \cdot p/2}{(6,3 - 6\frac{\%incid}{100}) \cdot (8 * Cos(\lambda^2 \pi) - Sin(\lambda \pi) + 11)} \tag{9}$$

This formula is more accurate for λ values close to 0 (from 0,6 to 0), due to the fact that the energy consumption is more considered for these values in the objective function. Thus given a percentage of robustness for a given incidence duration (%incid) and a λ value of a schedule, we can estimate the energy needed to carry out this schedule. In the same way, a schedule with a given energy consumption, a λ value and a threshold of the duration of the incidences (%incid), we can estimate the robustness of this schedule. This formula can be refined by the operator according on the distribution of energy consumption of machines at different speeds.

9 Makespan versus Robustness

There is a direct relationship between Makespan and Robustness because as makespan increases, the robustness is bigger due to the fact that the tasks are more sparse in time and they are able to absorb more incidences. However it is not realistic to generate too sparse schedules, so generally a makespan bound is set and we try to find the more robust schedule for a given makespan threshold.

To carry out this study, the simulation carried out in the previous section gave us the amount of incidences that can be absorb by modifying the energy consumption threshold. Figure 4 shows an approximate Pareto Front for a set of 10 instances with 7 machines, 10 tasks per job and a maximum processing time of 100 time units. It can be observed that as the makespan increased the robustness also increased with a trigonometrical shape. Table 3 shows the makespan and robustness in different instances. It must be taken into account that the robustness is quite similar in all instances, so it is not directly dependent on the number of machines, number of tasks per job, neither the range of processing times. When the makespan threshold was set to the minimum possible (to achieve the optimal solution), these solutions were able to absorb an average of 29% of the incidences (first row of Table 3. This is due to the fact that natural buffers(Black rectangles in Figure 1) were able to absorb this percentage of incidences. Finally, when the makespan threshold set to an upper bound (obtained by minimizing energy consumption), the percentage of absorbed incidences was close to 100%. That means that the buffers are well distributed among all tasks and almost all disruptions were able to be absorbed.

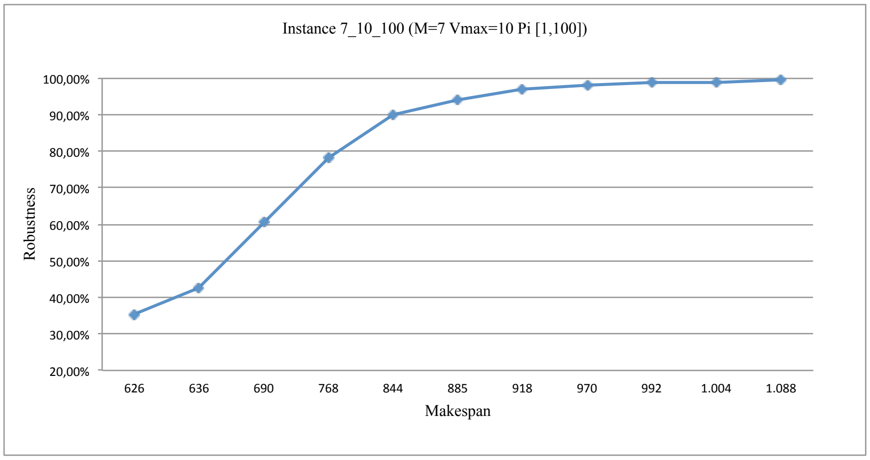

Figure 4: Approximate Pareto Front for the bicriteria Makespan-Robustness.

Table 3: Makespan and Robustness in instances $< m, v_{max}, p >$

	5_10_50		**7_10_100**		**3_20_50**		**3_25_100**		**3_30_200**	
λ	**Mk**	**Robust.**	**Mk**	**Robust.**	**Mk**	**Robust.**	**Mk**	**Robust.**	**Mk**	**Robust.**
1	317	26.7%	625.9	29.2%	707.8	26.8%	1686.1	27.0%	4048.9	25.6%
0,9	322.8	37.2%	635.7	37.9%	728.3	36.3%	1738.4	36.8%	4182.9	39.7%
0,8	343.1	48.2%	690.1	49.0%	777.8	50.6%	1854.4	49.5%	4530.8	48.9%
0,7	384.7	68.1%	768.2	67.5%	854.6	66.0%	2056.1	67.9%	4953.3	65.2%
0,6	410.1	79.8%	843.7	82.6%	946.8	79.6%	2280.6	78.9%	5452	80.9%
0,5	454.3	86.6%	884.5	89.1%	1011.7	88.0%	2421.8	86.3%	5825	88.0%
0,4	483.6	91.8%	918.3	89.8%	1079	93.4%	2553	94.1%	6158	92.7%
0,3	502.6	94.2%	970.2	94.8%	1112.6	95.3%	2667.1	94.9%	6364.8	95.9%
0,2	515.8	97.0%	992.1	94.3%	1145.7	96.6%	2734.2	96.0%	6513.2	96.8%
0,1	524.4	93.7%	1004.3	97.1%	1168.7	97.3%	2768.4	97.6%	6600.8	96.3%
0	565.4	97.2%	1088.4	96.2%	1296	99.0%	3160	98.9%	7289.6	99.2%

The relationship between makespan and robustness can be obtained from the above formulas (8) and (9) to obtain formula (10):

$$\frac{Makespan(\lambda)}{Robustness(\lambda, \%incid)} \approx$$

$$\approx \frac{ttasks \cdot p/2}{(6,3 - 6\frac{\%incid}{100}) \cdot (8 * Cos(\lambda^2\pi) - Sin(\lambda\pi) + 11) \cdot (e^{(0,25+1,2\lambda)} - \frac{Sin(\lambda\pi)}{2} + \frac{Cos(\lambda\pi)}{8})} \qquad (10)$$

Thus, given a makespan of a schedule with a given λ value, and the duration of the incidence (%incid), we can estimate the robustness of this schedule. In the same way, given a robustness threshold, the duration of the incidence (%incid) and a λ value, we can estimate the makespan of this schedule. This formula can be refined by the operator according on the distribution of energy consumption of machines at different speeds.

10 General Analysis

In this section, a general analysis for all instance types was carried out. The main objective is to analyze the relationship among all relevant parameters around robustness and energy efficiency for all analyzed instances and different λ value (horizontal axe). Figure 5 shows the results for disruptions of 40% of the maximum processing time (%incid=40). The main vertical axe represents the robustness. Thus, the blue curve (*% of Absorbed (40%)*) represents the percentage of absorbed incidences for each λ value. The yellow curve (*%Natural Buff*) represents the percentage of incidences absorbed by a natural buffer. The green curve (*%EffEn Buff*) represents the percentage incidences absorbed by accelerating a machine. In this way the robustness is *% of Absorbed (40%)=%Natural Buff+%EffEn Buff.*

In the secondary vertical axe, the garnet columns (*NbuffEff*) represent the average number of buffers generated by increasing the speed of machines and light blue columns (*NbuffNat*) represents the average number of natural buffers.

It can be observed that *NbuffNat* is mainly constant because they are independent of the objective (minimize makespan or energy consumption). However the total amount of time involved in these natural buffers decreased as the value of λ increased. This

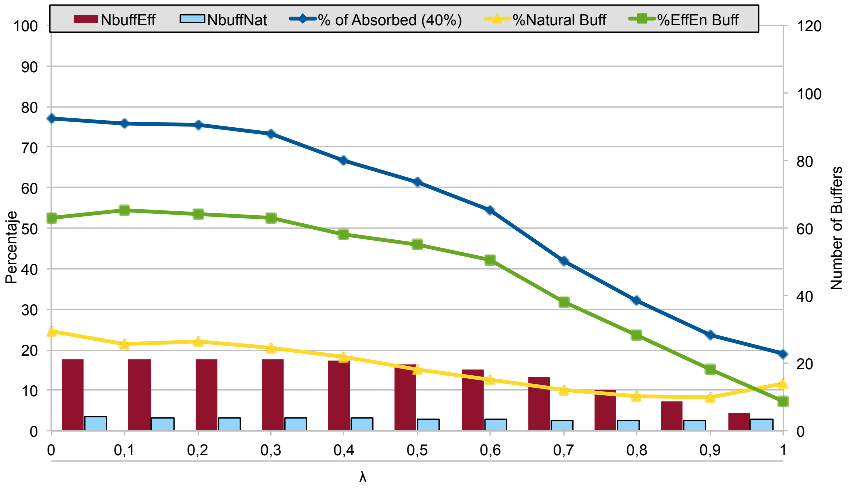

Figure 5: General Analysis for disruptions of 40% of maximum processing time (%incid=40).

is due to the fact that as λ increased the objective function give more importance to minimize makespan, so the free slack is also minimized. Thus, the percentage of times that the incidence is absorbed by a natural buffer (*%Natural Buff*) also decreased. The same tendency is carried out by *NbuffEff* where the number of buffers generated decreased as λ increased. This is due to the fact that, as λ increases, the objective is to minimize makespan and more machines are assigned at maximum speed, so few buffer times can be generated by speeding up the machines. The percentage absorbed incidences (*%EffEn Buff*) also decreased as the λ increased. However, it can be observed the difference between the percentage of absorbed incidences by the speeding up the machines (*%EffEn Buff*) against the percentage of absorbed incidences by natural buffers (*%Natural Buff*). Indeed the main objective is represented by the blue curve (*% of Absorbed (40%)*) that represents the percentage of absorbed incidences for each λ value. It can be observed that for $\lambda = 0$ (minimizing energy consumption) almost all incidences can be absorbed. Thus, energy-aware schedules are also considered robust solutions that can absorb medium-size incidences.

In Figure 6 we have simulated disruptions of different length, from 10% to 40% of the maximum processing time (from %indic=10 to %indic=40). The red curve (*% of absorbed (10%)*) represents the percentage of absorbed incidences for each λ value. The green curve (*% of absorbed (20%)*) represents the percentage of absorbed incidences for each λ value. The garnet curve (*% of absorbed (30%)*) represents the percentage of absorbed incidences for each λ value. Finally, the grey curve (*% of absorbed (40%)*) represents the percentage of absorbed incidences for each λ value. It can be observed that all curves maintained the same behavior in all λ values and the values are proportional to the length of the disruptions. This is due to the fact that it is easier to absorb small incidences that higher, but the difference is not too high. Thus, longer

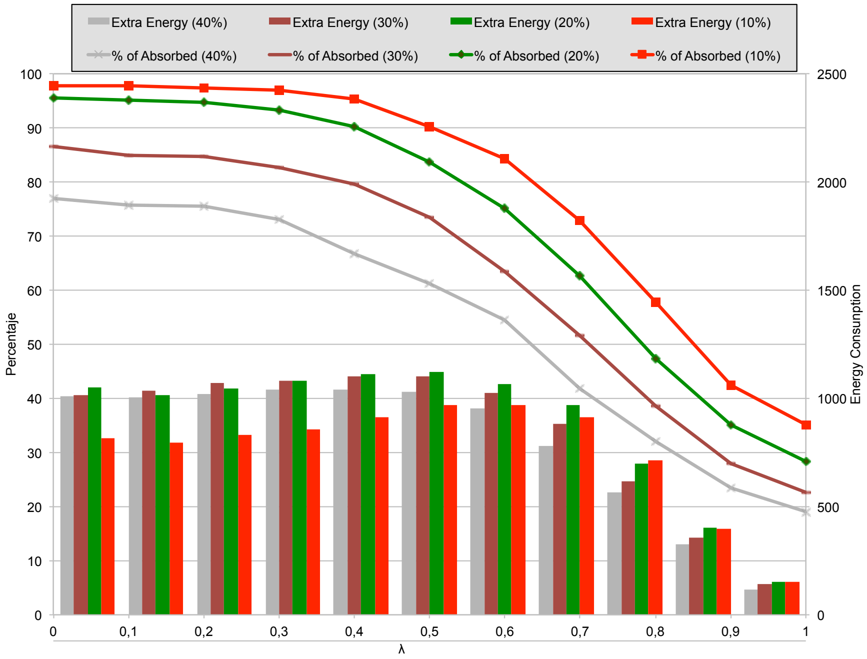

Figure 6: Disruptions from 10% to 40% of maximum processing time.

incidences than 40% will maintain the same tendency (proportional to the presented in Figure 6).

In the secondary vertical axe, the extra energy needed to absorb incidences is represented for the different length of disruptions (from 10% to 40%). It can be observed that although *% of Absorbed (20%)* was able to absorb less disruptions than *% of Absorbed (10%)*, it needed more extra energy than the other in many cases. It must taken into account that as the number of absorbed disruptions increased, the extra energy needed to absorb these disruptions also increased and the magnitude of needed energy is proportional to the size of the disruption. For instance, for $\lambda = 0$ (minimizing energy), the percentage of absorbed disruption of size 40% was around 77%, meanwhile the percentage of absorbed disruption of size 30% was around 87%. However, the extra energy needed to absorb these incidences was almost the same in both cases, due to the fact that larger disruptions generated larger needed of extra energy.

11 Conclusions

Many real life problems can be modeled as a job-shop scheduling problem where machines can work at different speeds. It represents an extension of the classical job-shop scheduling problem, where each operation has to be executed by one machine and this machine has the possibility to work at different speeds. In this paper, we analyze the relationship among three important objectives that must be taken into consideration:

Energy-efficiency, Robustness and Makespan. Analytical formulas are presented to estimated the relationship between these objectives in the analyzed instances. The results show the tradeoff between makespan and robustness, and the direct relationship between robustness and energy-efficiency.

To reduce the makespan, the energy consumption has to be increased to process the tasks faster. When the energy consumption is low, it is due to the fact that the machines are not working at highest speed so if an incidence occurs, the speed of these machines can been increased in order to recover the time lost generated by the incidence. So robustness is directly related with energy consumption. Robustness is also directly related with makespan because when makespan increases, there are more gaps in the solution, so sometimes incidences can be absorbed by these natural buffers.

Thus, new techniques can be developed to find robust solutions and at the same time they are guaranteed to be energy-aware solutions. Thus, in online scheduling, the obtained robust solution is carried out and only in case of disruptions, the involved machines are accelerated to absorb the disruptions and the rest of the tasks are executed in an energy-aware scheduling.

In further works, we will develop new metaheuristic techniques for finding robust and energy-aware solutions. These problems have multiple objectives so efficient techniques must be developed to obtain optimized solutions in an efficient way.

Acknowledgments

This research has been supported by the Spanish Government under research project MICINN TIN2013-46511-C2-1-P and the CASES project supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme under the grant agreement No 294931. This research was also supported by National Science Foundation of China (No. 51175262) and Jiangsu Province Industry-Academy-Research Grant (No. BY201220116).

References

[1] A. Agnetis, M. Flamini, G. Nicosia, and A. Pacifici. A job-shop problem with one additional resource type. *Journal of Scheduling*, 14(3):225–237, 2011.

[2] JC. Billaut, A. Moukrim, and E. Sanlaville. Flexibility and robustness in scheduling. Wiley, 2008.

[3] J. Blazewicz, W. Cellary, R. Slowinski, and J. Weglarz. Scheduling under resource constraints-deterministic models. *Annals of Operations Research*, 7:1–356, 1986.

[4] BMWi. German federal ministry of economics and technology: Energy statistics. 2009.

[5] AAG Bruzzone, D. Anghinolfi, M. Paolucci, and F. Tonelli. Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops. *CIRP Annals-Manufacturing Technology*, 2012.

[6] A. Caplinskas, G. Dzemyda, F. Kiss, and A. Lupeikiene. Processing of undesirable business events in advanced production planning systems. *Informatica: International Journal*, 23(4):563–579, 2012.

[7] J. Dahmus and T. Gutowski. An environmental analysis of machining. In *ASME International Mechanical Engineering Congress and RD&D Exposition, Anaheim, California, USA*, 2004.

[8] M. Dai, D. Tang, A. Giret, M.A. Salido, and W.D. Li. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, to appear, 2013.

[9] J.R. Duflou, J.W. Sutherland, D. Dornfeld, C. Herrmann, J. Jeswiet, S. Kara, M. Hauschild, and K. Kellens. Towards energy and resource efficient manufacturing: A processes and systems approach. *CIRP Annals-Manufacturing Technology*, 2012.

[10] K. Fang, N. Uhan, F. Zhao, and J.W. Sutherland. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, 30(4):234–240, 2011.

[11] A. Garrido, M.A. Salido, F. Barber, and MA López. Heuristic methods for solving job-shop scheduling problems. In *ECAI-2000 Workshop on New Results in Planning, Scheduling and Design, Berlín*, pages 36–43, 2000.

[12] T. Gutowski, C. Murphy, D. Allen, D. Bauer, B. Bras, T. Piwonka, P. Sheng, J. Sutherland, D. Thurston, and E. Wolff. Environmentally benign manufacturing: observations from japan, europe and the united states. *Journal of Cleaner Production*, 13(1):1–17, 2005.

[13] K.L. Huang and C.J. Liao. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & Operations Research*, 35(4):1030–1046, 2008.

[14] IBM. Modeling with IBM ILOG CP Optimizer - practical scheduling examples. *IBM*, 2007.

[15] IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[16] L. Kramer, L. Barbulescu, and S. Smith. Understanding performance tradeoffs in algorithms for solving oversubscribed scheduling. In *Proceedings 22nd Conference on Artificial Intelligence (AAAI-07)*, July 2007.

[17] P. Laborie. IBM ILOG CP Optimizer for detailed scheduling illustrated on three problems. In *Proceedings of the 6th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR09)*, pages 148–162, 2009.

[18] W. Li, A. Zein, S. Kara, and C. Herrmann. An investigation into fixed energy consumption of machine tools. *Glocalized Solutions for Sustainability in Manufacturing*, pages 268–273, 2011.

[19] G. Mouzon and M.B. Yildirim. A framework to minimise total energy consumption and total tardiness on a single machine. *International Journal of Sustainable Engineering*, 1(2):105–116, 2008.

[20] G. Mouzon, M.B. Yildirim, and J. Twomey. Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, 45(18-19):4247–4271, 2007.

[21] R. Neugebauer, M. Wabner, H. Rentzsch, and S. Ihlenfeldt. Structure principles of energy efficient machine tools. *CIRP Journal of Manufacturing Science and Technology*, 4(2):136–147, 2011.

[22] E Nowicki and C Smutnicki. An advanced tabu search algorithm for the job shop problem. *Journal of Scheduling*, 8:145–159, 2005.

[23] Y. Seow and S. Rahimifard. A framework for modelling energy consumption within manufacturing systems. *CIRP Journal of Manufacturing Science and Technology*, 4(3):258–264, 2011.

[24] E. Szathmary. A robust approach. *Nature*, 439:19–20, 2006.

[25] G. Verfaillie and T. Schiex. Solution reuse in dynamic constraint satisfaction problems. In *Proc. of the 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 307–312, 1994.

[26] N. Weinert, S. Chiotellis, and G. Seliger. Methodology for planning and operating energy-efficient production systems. *CIRP Annals-Manufacturing Technology*, 60(1):41–44, 2011.