A Real Delivery Problem Dealt with Monte Carlo Techniques

P. Fernández de Córdoba, L.M. García-Raffi, A. Mayado and J.M. Sanchis
Departamento de Matemática Aplicada.
Universidad Politécnica de Valencia, E-46071 Valencia, Spain
jmsanchis@mat.upv.es

Abstract

In this paper we use Monte Carlo Techniques to deal with a real world delivery problem of a food company in Valencia (Spain). The problem is modeled as a set of 11 instances of the well known Vehicle Routing Problem, VRP, with additional time constraints. Given that VRP is a NP-hard problem, a heuristic algorithm, based on Monte Carlo techniques, is implemented. The solution proposed by this heuristic algorithm reaches distance and money savings of about 20% and 5% respectively.

Key= Words: Vehicle Routing Problem; Time Windows; Monte Carlo Methods; Heuristic Algorithms;
AMS= subject classification: 1207.04, 1203.15, 1203.26

1 Introduction

In this paper we describe our experience in optimizing the delivery routes of a well known food company in Valencia (Spain), starting at a huge store located at 13 km. from the city and serving up to 70 food shops located inside the city. Given that several vehicles are needed to perform the delivery, this problem can be modeled as the *Vehicle Routing Problem(VRP)*, that we define next.

Let $G = (V, E)$ be a graph where $V = \{1, 2, \ldots, n\}$ is a set of vertices representing shops with the *depot* located at vertex 1, and E is the set of edges. Let c_{ij} be a non-negative *travel cost* associated to each edge (i, j) and let d_i be a non-negative *demand* associated to each vertex $i > 1$. Finally, let assume that there is available a number of *vehicles* with equal capacity Q. The *VRP* consists of designing a set of minimum cost vehicle routing satisfying:

This work has been partially supported by the *Plan de Incentivo a la Investigación/98* of the Universidad Politécnica de Valencia, under the project *"Técnicas Monte Carlo aplicadas a Problemas de Rutas de Vehículos"*.

(i) each vertex in $V \setminus \{1\}$ is visited by exactly one vehicle exactly once.
(ii) all vehicle routes start and end at the depot.
(iii) some additional constraints are satisfied.

The most usual additional constrains include:

(a) *Capacity constraints:* the sum of demands of any vehicle route cannot exceed the vehicle capacity. Capacity-constrained VRPs are referred to as CVRPs.
(b) *Total time constraints:* the length of any vehicle route cannot exceed a prescribed bound T, where the c_{ij} are considered as travel times and the d_i as stopping times (for service) at each shop i. Distance —or Time— constrained VRPs are referred to as DVRPs.

A wide variety of exact and heuristic algorithms have been proposed for the VRP (see the Laporte and Nobert (1987) survey and the Laporte (1992) review, for example). Exact algorithms for the VRP are based in direct tree search methods, dynamic programming or integer linear programming. Heuristic algorithms are derived from procedures for the TSP, ensuring that only feasible routes are created.

Most of these algorithms are designed to be applied —with small changes— either to CVRP or to DVRP. There are also approaches to deal with VRP facing both Capacity and Distance constraints (see Laporte et al., 1985). To our knowledge, in all these models there is a one-to-one correspondence between the set of vehicles and the set of single routes —all starting at the depot, serving a set of nodes and ending at the depot—. This feature is absent in the problem we face in this work, since the capacity constraint here is much more restrictive than the time constraint and a given vehicle must perform several single travels (see figure 2 and table 2). Each single travel is limited by the capacity constraint, and the set of travels performed by a given vehicle is limited by the time constraint. More exactly, there are two non-negative numbers c_{ij} and t_{ij} associated with each edge (i, j) (travel costs and travel times, respectively), two non-negative numbers d_i and t_i associated with each vertex i (demand and stopping time, respectively), a capacity Q for each vehicle and a time bound T for the length of each vehicle routing. The demands of the nodes served in a single travel must not exceed the vehicle capacity Q. The sum of the times spent by all the travels performed by a vehicle must not exceed the time bound T.

It should be noticed that ignoring the time bound T —it could be an appealing approach to our problem because of the relative importance of the two constraints— and solving the problem as a pure CVRP, we would obtain single routes much shorter than the time bound. After that, these routes should be clustered into groups to be performed by a single vehicle, in such a way that the sum of the times estimated for each vehicle does not exceed the time bound T. This approach could provide inappropriate solutions (see the example on section 3). In this sense, existing algorithms for VRP are not easy to apply to our problem and we design a specific algorithm based on Monte Carlo (MC) techniques.

The features of the real delivery problem are presented in section 2. In section 3 we formulate the problem as a DCVRP which is dealt with a MC algorithm described in section 4. Computational results and the corresponding estimation of savings are shown in section 5 and conclusions are given in section 6.

2 Brief description of the real situation.

As it was mentioned before, the food company has a huge store located at 13 Km. from Valencia and must serve up to 70 food shops located inside the city. Figure 1 shows the location of each shop in the map of Valencia. The main store is not represented. The depot of the VRP instances (represented by a square) stands for the point where the route arrives to the city from the main store.

The agreement between the shops and the food company establishes that each shop must demand a fixed number of pallets in certain days of the week. For example, shop #1 demands 3 pallets every monday morning. Figure 2 shows the 27 shops having demand every monday morning. The contents of each pallet can be different from week to week, but the number of pallets is fixed. Thus, the unit of demand and transportation is the pallet. The number of pallets demanded by a shop ranges from 3 to 23, with an average of 8.64 pallets and the vehicles used have a capacity of 23 pallets. Finally, the schedule-times of the food shops in Valencia implies that pallets must be served within a given time window, from 7 to 12 hours in the morning and from 15 to 20 hours in the afternoon.

Due to the nature of the demand of each shop, the company has de-

fined 11 'delivery problems': monday-morning (showed in Figure 2 and in table 2), monday-afternoon, thursday-morning, ..., friday-afternoon and saturday-morning. In each delivery problem up to 27 shops have to be served by a set of vehicles (trucks) with the same capacity $Q = 23$ pallets. Each vehicle performs up to 4 different travels to the shops, all starting and ending at the depot and with the sum of demands not exceeding 23. These travels are determined by the constraint that all the shops must be served within the time window.

We face a *double objective*. First, our aim is to minimize the number of vehicles used. Second, we seek to minimize the total length covered by this minimum set of vehicles.

3 Problem formulation.

Fixed means here that the subproblem monday-morning (for example) is exactly the same for every monday of the year, i.e., we do not have to solve an instance with different data for every monday. Once the 11 instances corresponding to the 11 given subproblems have been solved, the solutions proposed will be valid for the whole year. In order to define each subproblem exactly, let us define:

subroute: single travel starting and ending at the depot that serves several shops whose total demand is no greater than Q.

route: set of *subroutes* (performed by a single vehicle) such that the total time needed to complete them does not exceed T.

We have to design a minimum set of routes formed by subroutes serving all the shops with minimum total length. Hence, the 11 instances, with size ranging from 12 to 27 nodes, can be considered as instances for the DCVRP. The following data are required for each instance:

1. List of shops with their number d_k of pallets demanded.
2. Distance matrix c_{ij} among every pair of nodes i and j (shops and depot).

3. Vehicle capacity Q (fixed to 23 pallets).
4. Total time bound T (fixed to 360 minutes).
5. Time matrix t_{ij} needed to travel from node i to node j.
6. Time t_0 needed to load a vehicle at the depot (including the time needed to go and come from the city to the main store).
7. Time t_k needed to unload d_k pallets at the shop k.

In order to obtain the previous data concerning time (4 to 7), we collect the time spent by 11 real vehicles in their delivery routes (selecting one truck in each subproblem) and we estimate:

- an average speed of 21 Km/h for a truck traveling on the streets of Valencia. Hence, we can compute times t_{ij} from lengths c_{ij}.
- an average time of 1 minute to load/unload one pallet from a truck. Hence, we can compute times t_k from demands d_k.
- a time of 10 minutes needed to travel from the main store to the city (13 Km.). Hence, we can compute the time t_0 and the total time bound, $T = 360$ minutes: 5 hours corresponding to the schedule-times of the shops plus the time estimated to load a vehicle at the store, to travel from the store to the first shop served and to return from the last shop served to the store.

The features of these data make capacity constraints more restrictive than time constraints. A single travel is limited by the capacity constraint $Q = 23$, but several single travels can be performed by the same vehicle without violating the time bound $T = 360$ minutes. Nevertheless, as we mentioned in section 1, we can not ignore the time bound T and solve the instances as pure CVRP, because we would not obtain good solutions. Consider, for example, the first instance to be solved, corresponding to monday-morning. Solving this instance as a pure CVRP, we obtained the solution presented in table 1, with 9 single travels and 368222 metres long. The last row of table 1 presents the estimated time needed to perform each single travel. Given that the time bound is $T = 360$ minutes, these 9 single travels can not be performed by only 4 vehicles: in such a case, one vehicle would be forced to perform 3 single travels, with an estimated time of, at least, 386.2

Single travel	1	2	3	4	5	6	7	8	9
Pallets served	22	22	22	21	23	20	23	22	23
Time (min.)	129.4	130.7	126.1	144.4	151.8	137.9	137.2	142.9	138.1

Table 1: Solution to Mon-m as pure CVRP. Total cost: $z = 368222$ m.

minutes —corresponding to travels 1, 2 and 3—. Nevertheless, when we solve this instance by considering the time bound $T = 360$, we obtain the solution in table 2 —represented in figure 2— also with 9 single travels. This solution has a total length of 373069 metres, higher than the previous solution, but it can be performed by only 4 vehicles without violating the time bound T.

Vehicle	1	1	1	2	2	3	3	4	4
Single travel	1	2	3	4	5	6	7	8	9
Pallets served	22	20	19	22	23	23	23	23	23
Time (min.)	137.9	109.9	107.3	140.0	158.6	152.3	137.2	162.0	147.3

Table 2: Solution to Mon-m with time bound $T = 360$. Total cost: $z = 373069$ m.

4 The MC algorithm.

As mentioned in the Introduction, VRP is a $\mathcal{NP}$-hard problem and therefore in order to face it we implement a heuristic algorithm, based on Monte Carlo techniques. These techniques applied to Routing Problems were first introduced by Fernández de Córdoba et al. (1998) for the Rural Postman Problem. We simulate a vehicle traveling randomly over the edges of a graph, describing a tour by jumping from one node to another depending on certain probabilities. The key of these models is the definition of the probabilities that depend on the tours to be designed. In this case, probabilities are defined to design *subroutes* —serving nodes with total demand not exceeding Q— and *routes* —formed by a set of subroutes with total accumulated time not exceeding T—. When all the nodes have been served, the vehicle ends at the depot. A number of tours (iterations) is tried, and the best is selected as the output of the algorithm.

At the beginning of each iteration we consider the *subroute #1* of the

route #1, and we assume the vehicle is at the depot. We also set

$$\begin{array}{c}\text{length: } z = 0\\ \text{load: } q = 0\\ \text{time: } t = t_0\end{array}$$

Each *subroute* is stored in an integer vector with a record for each edge in the graph. Every time an edge (i, j) is traversed, we increment by one the corresponding record and we update:

$$\begin{array}{c}\text{length: } z = z + c_{ij}\\ \text{time: } t = t + t_{ij}\end{array}$$

and every time a node k is reached and served we label it as 'served by the current *subroute*' and we update:

$$\begin{array}{c}\text{load: } q = q + d_k\\ \text{time: } t = t + t_k\end{array}$$

Suppose now the vehicle is at node i. Probability p_{ij} of traversing edge (i, j) must depend on the cost c_{ij}, on the value of demand d_j versus the current load capacity $Q-q$, and on the time $t_{ij}+t_j+t_{j1}$ versus the remaining available time $T - t$. Based on our experience in MC techniques (see Fernández et al. 1998,1999a, 1999b), the probabilities have been modelled in the following way:

(a) select nodes j such that $q+d_j \leq Q$ (nodes j that can still be served) and such that $t + t_{ij} + t_j + t_{j1} \leq T$ (nodes j that can still be visited within the schedule-time).

(b) If there are such nodes, select one of them with probabilities p_{ij} proportional to $1/c_{ij}^{\alpha}$, where α is a real parameter.

(c) If it does not exist a node j such that $q + d_j \leq Q$, i.e., if any node j can be served, then the vehicle returns to the depot and we start a new *subroute*, setting the load $q = 0$ and $t = t + t_0$. If it does not exist a node j such that $t + t_{ij} + t_j + t_{j0} \leq T$, i.e., if any node j can be visited and served within the schedule-time, then the vehicle returns to the depot and we start a new *route* —and a new *subroute*—, setting the load $q = 0$ and $t = t_0$.

Finally, when all the nodes have been served, the vehicle returns to the depot and we have a possible routing solution. The algorithm generates one possible solution in each iteration. Among these solutions, the one satisfying that, first, it uses the minimum number of vehicles (*routes*) and, second, its travelled length z is minimum, is proposed as the output of the algorithm.

The real parameter α increases the difference among the probabilities of different edges. A greater value of α means greater differences among the p_{ij}'s corresponding to edges (i, j) with different costs c_{ij}. A very large value for α could be understood as a 'greedy' algorithm in the sense that the shortest edge is usually selected. On the other hand, a value $\alpha = 0$ could be understood as a completely random algorithm. In order to check the influence of the value for α in the goodness of the solution obtained, we executed the MC algorithm for the 11 instances and for different values of α. The number of iterations used, 75000, is enough to provide significative results keeping low running times —ranging from 15 to 70 seconds for each instance—. Figure 3 shows, for the 11 instances, the global improvement obtained by the MC algorithm with different values for α, in percentage with respect to the current solution. The best results where obtained with $\alpha = 4$ (19.8% of improvement). It is worthwhile to notice that the worse results were obtained with values $\alpha = 0$ and $\alpha = 10$.

It could be argued that the effect of fixing the value of a parameter is to drive the search of solutions to a given (supposed good) zone of the whole search space. However, our goal in this problem is to obtain solutions as good as possible, by running the algorithm for a large number of iterations. Notice that we are not concerned about having a large running time because we do not have to solve fastly a different instance every day. Then, fixing the value of α could mean that we explore in excess a given zone and do not explore others, and it could be more efficient to diversify the zones to explore. Therefore, we also executed the MC algorithm with the value for α randomly changing after a given number of iterations. The parameter, initially fixed at $\alpha = 4$, is changed every 1000 iterations (for example) by adding a random real number ranging in $(-1, 1)$ (when this sum exceeds the interval $(0, 10)$, we set $\alpha = 4$ again).

The results obtained with α variable, also showed in figure 3, improves the result obtained with any fixed value. Hence, the algorithm we decide to run is the MC algorithm with α variable, for a very large number of

iterations —4000000—.

5 Computational Results.

4000000 iterations to the 11 instances on a Intel PC with a Pentium II 400 Mhz processor. The total time used for solving the 11 instances has been about 7 hours and 22 minutes. Table 3 shows, for each instance, the length, number of *routes* (vehicles) and number of *subroutes* (single travels) used by the solution currently implemented by the company and by the solution reported by our algorithm. It also shows the length savings in Km. and in percentage and the running time in seconds. The total length traveled by the trucks in the current solution is reduced by 962 km. each week. This represents an improvement of 20.75%. Notice that, with 75000 iterations, the algorithm reaches an improvement of 19.90%, with a running time inferior to 1 minute per instance. Therefore, it seems unlikely to obtain a much better solution by increasing the number —4000000— of iterations.

In order to estimate money savings we discuss briefly the way the company computes transportation costs. Instead of owning trucks, the company hires routeers with their own route on a day-to-day basis. The price paid by the company to each routeer depends on the number of single travels and km. covered and on the total value (invoice) of the merchandise carried. The number of travels and km. covered by each truck were known. The value of the carried merchandise in each travel varies from week to week, and it had to be estimated from the collected available data of the company (for details, see Mayado, 1998). These data are reported in table 4. For each day of the week, the (averaged estimated) cost of both the current solution and the solution proposed by the MC algorithm, the savings in 'pesetas' and the percentages are shown. Multiplying the (averaged) saving per week by 53 weeks per year, a total saving of 3.534.358 'pesetas'. (about 25.000 U.S. $) is obtained.

6 Conclusions.

In this paper we deal with a real world delivery problem of a food company in Valencia (Spain). The problem is modelled as a set of 11 instances of

	# shops	Current solution length (m.)	Current solution vehic.& travels	Proposed solution length (m.)	Proposed solution vehic.& travels	Savings Km.	Savings (%)	run. time (s)
mon-m	27	553129	5 & 14	373069	4 & 9	180	32.6	3689
mon-a	27	536868	5 & 14	429238	5 & 11	107	20.0	3736
tue-m	12	212306	2 & 5	176997	2 & 4	35	16.6	801
tue-a	20	386658	4 & 10	351185	4 & 9	35	9.2	2128
wed-m	21	369013	4 & 9	284585	3 & 7	84	22.9	2272
wed-a	25	510731	5 & 13	423876	4 & 11	86	17.0	3231
thu-m	15	282403	3 & 7	213288	3 & 5	69	24.5	1212
thu-a	21	422333	4 & 11	374816	4 & 10	47	11.3	2333
fri-m	23	370748	3 & 9	319419	4 & 8	51	13.9	2704
fri-a	26	608443	5 & 16	423983	4 & 11	184	30.3	3483
sat	13	384645	4 & 10	304776	3 & 8	79	20.8	984
Total a week		4637377	44 & 118	3675232	40 & 93	962	20.7	26528

Table 3: Length comparative data

the well known Vehicle Routing Problem, VRP, facing both Capacity and Time constraints. Each vehicle performs several travels within a given time interval. Each single travel —starting and ending at the depot— is limited by the capacity of the vehicle. To our knowledge, an algorithm to deal with this particular problem has not been proposed yet. We present a heuristic algorithm based on Monte Carlo techniques that has produced satisfactory computational results, obtaining distance and money savings of about 20% and 5% respectively.

Thus, we show that MC algorithms, previously studied by Fernández de Córdoba et al. (1998), are useful in implementing heuristic algorithms for different *Routing Problems*. The main advantage of MC algorithms is their simplicity (they are conceptually simple and easy to implement in a computer code) and their potential adaptability to a wide variety of situations. This feature has been essential to deal with a VRP problem with additional time constraints.

	current cost	MC sol. cost	savings (ptas.)	savings (%)
mon	287.477	274.417	13.060	4,54
tue	185.717	178.920	6.796	3,66
wed	231.801	224.282	7.519	3,24
thu	208.271	188.186	20.085	9,64
fri	284.373	266.334	18.039	6,34
sat	108.877	107.691	1.186	1,09
Total per week	1.306.516	1.239.830	66.686	5,01
Total per year			3.534.358	

Table 4: Economic comparative data

References

99

D=3D11 P. Fernández de Córdoba, L.M. García-Raffi and J.M. Sanchis Llopis (1998), "A heuristic algorithm based on Monte Carlo methods for the Rural Postman Problem". *Computers and Operations Research*, Vol. 25, No. 12, pp. 1097-1106, 1998.

D=3D12 P. Fernández de Córdoba, L.M. García-Raffi, E. Nieto and J.M. Sanchis Llopis (1999a), "Aplicación de técnicas Monte Carlo a un problema real de Rutas de Vehículos". *Anales de Ingeniería*, Colombia. In press.

D=3D1P. Fernández de Córdoba, L.M. García-Raffi and J.M. Sanchis Llopis (1999b), "A Constructive Parallel Algorithm based on Monte Carlo techniques for Routing Problems". Submitted to *Parallel Computers*.

D=3D1G. Laporte (1992), "The Vehicle Routing Problem: an overview of exact and approximate algorithms", *European Journal of Operations Research* 59, 345.

D=3D1G. Laporte, M. Desrochers and Y. Nobert (1985), "Optimal Routing under Capacity and Distance Restrictions". *Operations Research* 33, 1050-1073.

D=3D1G. Laporte and Y. Nobert (1987), "Exact algorithms for the The Vehicle Routing Problem". *Surveys in Combinatorial Optimization* (S. Martello, G. Laporte, M. Minoux and C. Ribeiro Eds.), North-Holland, Amsterdam.

D=3D1A. Mayado (1998), "Organización de los itinerarios de la flota de camiones de reparto de una sociedad cooperativa. Optimización mediante técnicas de simulación Monte Carlo". Proyecto Fin de Carrera. E.T.S.I.I, Universidad Politécnica de Valencia.

D=3D1

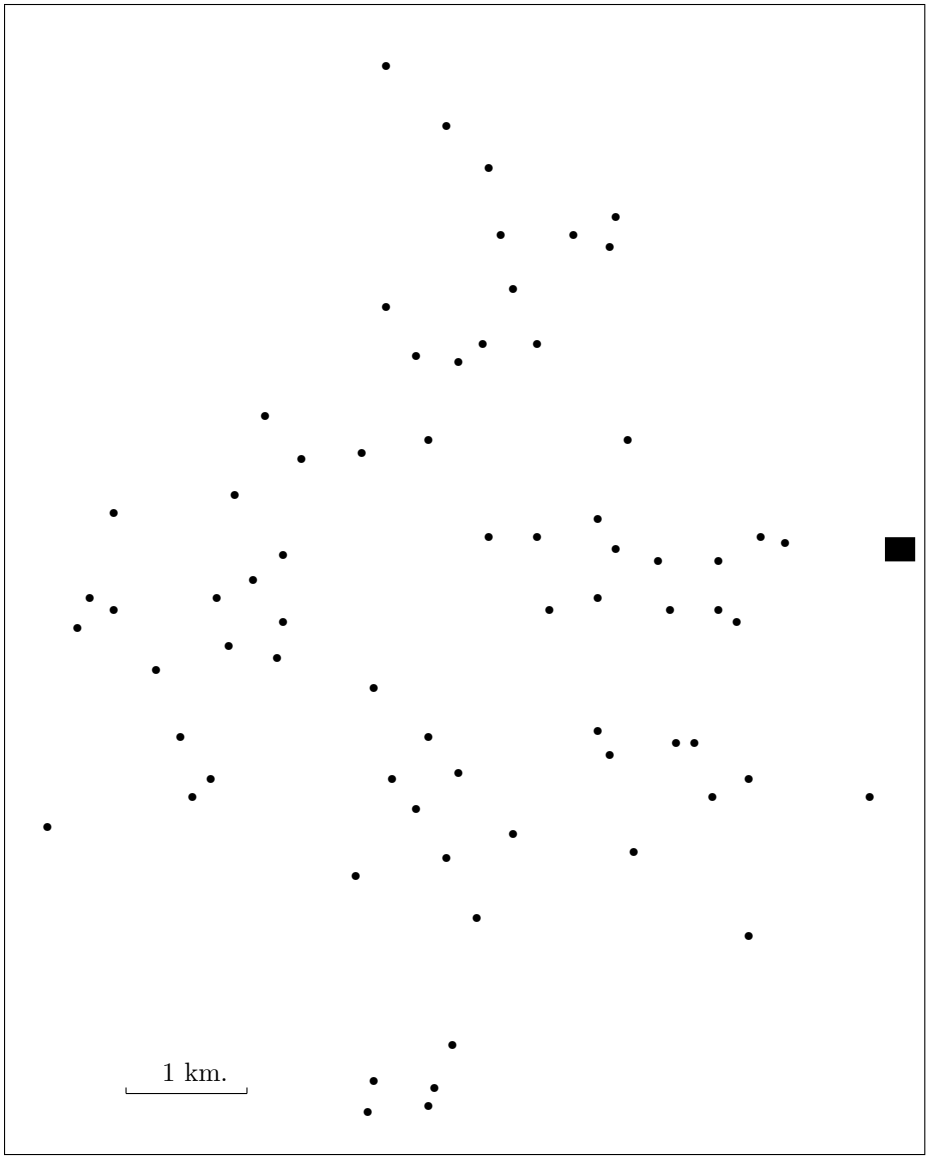

Figure 1: Location of the 70 food shops in Valencia.

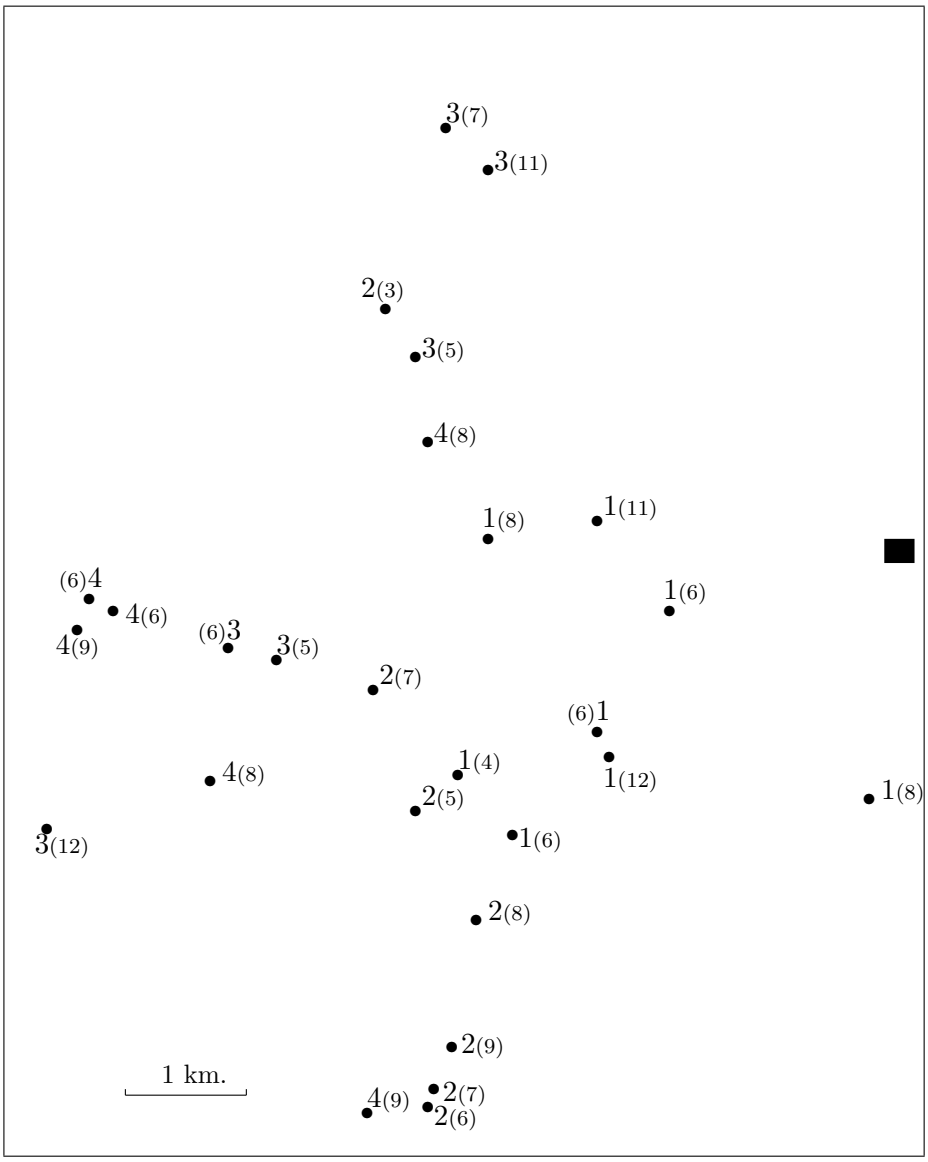

Figure 2: Location of the 27 shops having demand (in brackets) in monday-morning and solution provided by the MC algorithm. Each single travel is plotted except for the edges incident with the depot. Each shop is labeled with the vehicle number that serves it.

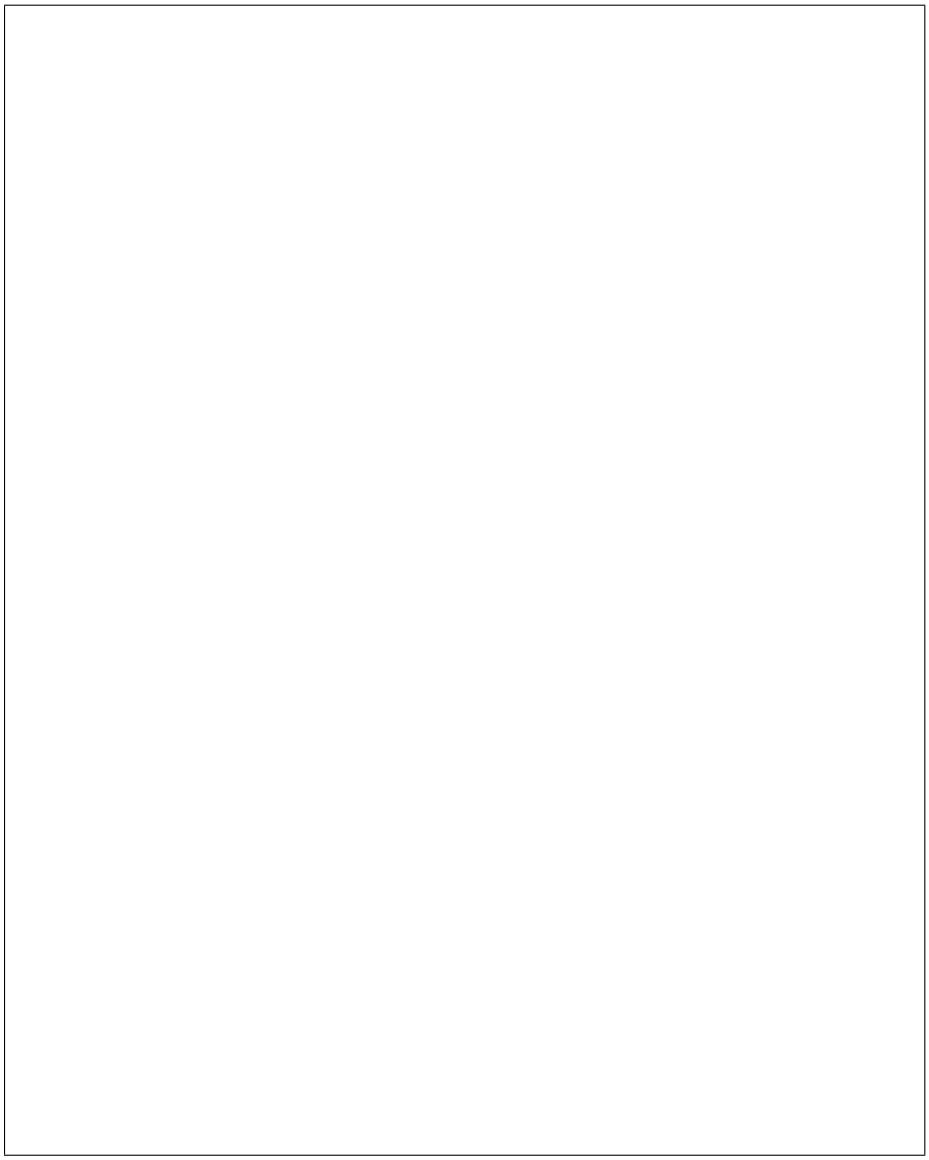

Figure 3: Percentage of global improvement, respect with the current solution, obtained by the MC algorithm with different uses for parameter α.