

A Heuristic Based Approach to Solve a Capacitated Location-routing Problem

Abbas Jokar¹ & Rashed Sahraeian¹

¹Department of Industrial Engineering, Shahed University, Tehran, Iran

Correspondence: Rashed Sahraeian, Department of Industrial Engineering, Shahed University, Persian Gulf Expressway, Tehran, Iran. Tel: 98-21-5121-2040. E-mail: sahraeian@shahed.ac.ir

Received: November 4, 2011 Accepted: December 12, 2011 Online Published: July 5, 2012

doi:10.5539/jms.v2n2p219 URL: <http://dx.doi.org/10.5539/jms.v2n2p219>

Abstract

In this paper we define a new approach to solve the location routing problem (LRP). LRP is a relatively new research direction within location analysis that takes into account vehicle routing aspects. In this research it is proposed a new heuristic method base on simulated annealing to solve a capacitated location routing problem. At first an initial solution is produced by a greedy approach and then encoding this initial solution to use in optimization language and finally apply neighborhood search method to improve the initial solution. Finally, to verify the quality of our new approach in term of solution quality, one well-known data problem set is selected. The numerical examples show the efficiency of our approach.

Keywords: location-routing problem, simulated annealing, neighborhood search method, greedy search method

1. Introduction

Today, companies, Due to excessive competition and Rapid advances in technology, are required to be neatly integrated into all manufacturing processes to be able to offer flexibility in their products. Supply chain management (SCM) as an integrated approach for the proper management of materials, information and financial flows, is able to meet these requirements. Location- routing problem (LRP) is considered one of the problems in distribution and support systems in supply chain.

Our definition consists of “LRP involves selecting optimal number of depots among several potential depots, allocate all customers to opened depots to service them by solving VRP with purpose of minimizing total cost, total cost includes open depot cost and routing cost, that consist of vehicle cost and travelling distance cost. In this optimization process two problems, selecting depot and vehicle routing, considers to improve simultaneously”. Without considering inter-relation between location and routing and solve these two problems separately, may cause enormous costs for the system.

Several methods have been introduced to solve such problems. All methods are divided into two categories: The exact methods and heuristic methods. Because of the LRP structure, apply the exact algorithm for solving the problem requires much time and money and in some cases when the problem size is large, the algorithm cannot find the optimum solution, therefore Most of the research to date has focused on heuristic methods since LRP merge two NP-hard problems.

The first exact algorithm for the general LRP is by Laporte and Nobert (1981); a branch-and-bound algorithm is used. Similar approach was applied to uncapacitated multifacility LRP by Laporte, Nobert, and Pelletier (1983) and capacitated multi-facility LRP by Laporte et al. (Laporte, Nobert, & Arpin, 1986). The latter applied a branching procedure where sub-tour elimination and chain barring constraints were reintroduced. Nagy and Salhi 2007 in a review article, to introduce almost the exact methods, had classified these methods.

Due to the exponential growth in the problem size, exact approaches for the LRP have been limited to small and medium size instances with 20–50 customers. For this reason, heuristics and meta-heuristics such as tabu search (TS), genetic algorithms (GA), simulated annealing (SA), and neural networks (NN), are often used to solve LRP instances in more recent studies. Wu et al, (2002) decompose the standard LRP with capacitated depots into a facility location-allocation Problem and a vehicle routing problem, and try then to solve both subproblems using simulated annealing. Melechovsky et al. (2005) address an LRP with nonlinear depot costs that grow with the total demand satisfied by the depots. They present a hybrid metaheuristic method consisting of tabu search

and variable neighborhood search heuristics. Marinakis and Marinaki (2008) introduced a hybrid metaheuristic for the LRP based on particle swarm optimization and path relinking. Duhamel et al (2008) proposed a memetic algorithm (genetic algorithm hybridized with a local search procedure; GAHLS) to the LRP.

Sahraeian and nadizadeh (2009) introduced a new method Using greedy clustering method to solve capacitated Location-routing problem in four phases. In phase 1, the customers are clustered according to greedy search method. Second phase, chooses the proper depot among candidates to be established. Third phase, allocates the clusters to depots. In final phase, Ant Colony System (ACS) is used for routing among depots and customers. In this paper we want to improve this greedy method to get better solution. According to, we introduce a new algorithm that apply SA method in it structure, improve neighborhood solution by use of 2-OPT, 3-OPT and Inverse 2-OPT, and last get a lot of attention to obtain initial solution.

There are two common optimal algorithms to solve LRP, exact algorithm and heuristics algorithm. In this paper by using simulated annealing (SA) method, which is one of the metaheuristic approaches to solve complex problems, we're trying to get the optimum solution or close to the optimal solution.

2. Mathematical Model

In this paper, we focus on solving the LRP with capacitated depots and routes. Prins et al. (2007) gives the following formal mathematical model for the problem.

Let $G = (V, E)$ be an undirected network where V is a set of nodes comprised of a subset I of m potential depot sites and a subset $J = V/I$ of n customers. E is a set of edges connecting each pair of nodes in V . Associated with each edge $(i, j) \in E$ is a traveling cost c_{ij} . Each depot site $i \in I$ has a capacity W_i and an opening cost O_i . Each customer $j \in J$ has a demand d_j which must be fulfilled by a single vehicle. A set K of identical vehicles with capacity Q is available. Each vehicle, when used by a depot i , incurs a depot dependent fixed cost F_i and performs a single route. Each route must start and terminate at the same depot, and its total load must not exceed vehicle capacity. The total load of the routes assigned to a depot must fit the capacity of the depot. The objective is to determine which depots should be opened and which routes should be constructed to minimize the total cost.

Define binary variables $y_i = 1$ iff depot i is opened, $f_{ij} = 1$ iff customer j is assigned to depot i , and $x_{jlk} = 1$ iff edge (j, l) is traversed from j to l in the route performed by vehicle $k \in K$. Then the problem can be formulated as the following binary integer program.

The objective function (1) is the sum of depot opening costs and the routing costs, including the travel costs and the fixed costs associated with vehicle uses. Constraints (2) ensure that each customer belongs to exactly one route, and that each customer has only one predecessor in the route. Constraints (3) and (4) are capacity constraints associated with routes and depots, respectively. Constraints (5) and (6) guarantee the continuity of each route, and that each route terminates at the depot where the route starts. Constraints (7) are sub-tour elimination constraints. Constraints (8) ensure that a customer must be assigned to a depot if there is a route connecting them. Finally, constraints (9), (10), and (11) specify the binary variables used in the formulation.

3. New Heuristic Method for Capacitated LRP

In the following subsections, we discuss the proposed heuristic method in detail, solution representation, the generation of the initial solution, various methods of finding neighborhood solution, and the new heuristic procedure.

3.1 Coding Solution

There are several methods for converting solution, of LRP problem, for use in program optimization such as MATLAB, GAMZ and etc. in this paper through new solution representation, which was introduced by yu et al (2010), apply coding solution to problem.

A solution is represented by a string of numbers consisting of a permutation of n customers denoted by the set $\{1, 2, \dots, n\}$, m potential depots denoted by the set $\{n + 1, n + 2, \dots, n + m\}$, and N dummy zeros which are used to separate routes, in addition to the vehicle capacity constraints. The i_{th} number in $\{1, 2, \dots, n\}$ denotes the i_{th} customer to be serviced. The first number in a solution is always in $\{n + 1, n + 2, \dots, n + m\}$ indicating the first depot under consideration. The solution representation is further explained as following. Each depot services customers between the depot and the next depot in the solution representation. The first route of this depot starts by servicing the first customer after the depot. Other customers for this depot are added to the current route one at a time. If adding a customer will exceed the vehicle's capacity, the current route is terminated. If the next number in the solution representation is a dummy zero, the current route will also be terminated. A new route

will be started to service remaining customers assigned to this depot.

Table 1. Customer location and demand

customer number	X	Y	W
1	7	6	23
2	8	8	25
3	5	6	10
4	4	1	30
5	7	6	26
6	11	9	14
7	12	5	14
8	10	4	31
9	2	4	22
10	1	7	13
11	9	6	15

Table 2. Depot location and capacity

depot number	X	Y	capacity
12	11	3	150
13	12	2	150
14	8	13	150
15	8	2	150

Tables 1 and 2 together give a small LRP instance with 11 customers and 4 potential depots. The location (X, Y) and demand (W) of customers are listed in Table 1. The location (X, Y) and capacity depot are given in Table 2. A randomly generated sample solution for this instance is shown in Figure 1. 3 dummy zeros are present in the solution. Figure 2 gives a visual illustration of the sample solution. In this example, the first number is depot 12, Customers (10, 9, 4, and 3) between depot 12 and depot 15 are real customers and thus depot 12 is opened to service these customers. The first route of depot 25 services customers 10, 9, 17 and 2. Customer 2 is followed by depot 15, thus the first route of depot 12 is terminated. No other routes are needed for depot 12. The first route of depot 15 services customers 1, 2, 5. Customer 5 is followed by a zero, so the first route is terminated. The second route of depot 15 then starts to service customers 11, 6, 7 and 8. Customer 8 is followed by depots 13 and 14 so the second route is terminated. At this point, all customers are serviced so no other depots need to be opened and the decoding process can be terminated.

$$\min = \sum_{i \in I} O_i y_i + \sum_{i \in V} \sum_{j \in J} \sum_{k \in K} C_{ij} X_{ijk} + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} F_i X_{ijk} \quad (1)$$

Subject to:

$$\sum_{k \in K} \sum_{i \in V} X_{ijk} = 1 \quad \forall j \in J \quad (2)$$

$$\sum_{j \in J} \sum_{i \in V} d_j X_{ijk} \leq Q \quad \forall k \in K \quad (3)$$

$$\sum_{j \in J} d_j f_{ij} \leq W_i y_i \quad \forall i \in I \quad (4)$$

$$\sum_{j \in V} X_{ijk} - \sum_{j \in V} X_{jik} = 0 \quad \forall k \in K, i \in V \quad (5)$$

$$\sum_{i \in I} \sum_{j \in J} X_{ijk} \leq 1 \quad \forall k \in K \quad (6)$$

$$\sum_{i \in S} \sum_{j \in S} X_{ijk} \leq |S| - 1 \quad \forall S \subseteq J, k \in K \quad (7)$$

$$\sum_{u \in J} X_{iuk} + \sum_{u \in V \setminus \{j\}} X_{ujk} \leq 1 + f_{ij} \quad \forall k \in K, j \in J, i \in I \quad (8)$$

$$X_{ijk} \in \{0,1\}, \quad \forall i \in I, j \in V, k \in K \quad (9)$$

$$y_i \in \{0,1\}, \quad \forall i \in I \quad (10)$$

$$f_{ij} \in \{0,1\}, \quad \forall i \in I, j \in V \quad (11)$$



Figure 1. An a example of coding solution

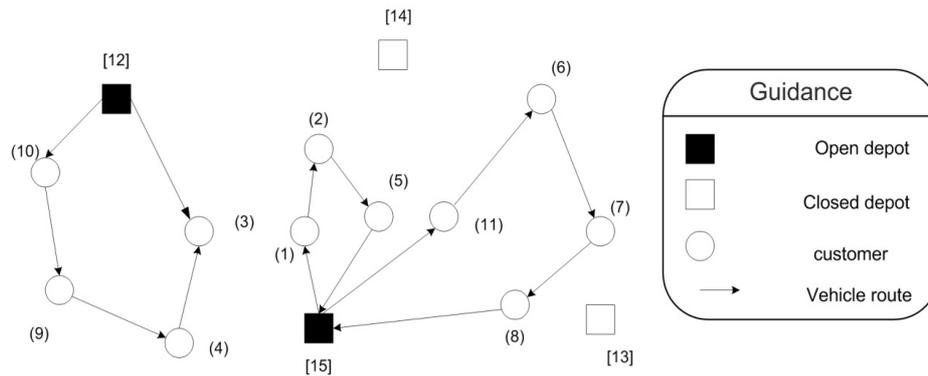


Figure 2. Visual illustration of the solution given in figure 1

3.2 Initial Solution

The initial solution is constructed by the following greedy method with the hope that a good initial solution can be found within a reasonable time.

Step 1

Let N be the set of unused depots. For each depot i in N , let $cc(i)$ be the number of unassigned customers whose closest depot in N is depot i . choose the depot in U with the highest cc value. If there is a tie, select the depot with the highest capacity. Go next step.

Step 2

For all unassigned customers, chose a customer that has closest distance to selected depot and start new tour and Go next step.

Step3

Among remain customer select that has minimum distance with the last assignment customer. If by adding new customer the capacity of tour was exceed, Go step 4 else add customer to tour and continue this step.

Step 4

Set $Re = \text{vehicle capacity} - \text{tour capacity}$. Customers that have demand Less than or equal (Re) put in list C . if C not empty, the member that has minimum distance add to tour and continue this step else go step 5.

Step 5

If sum demand of all assignment costumers was Less than or equal the capacity of selected depot, go to step 2 else go to step 6.

Step 6

Encode the current solution for this depot, then for $2000 * L$ times (L represents the length of solution encoding) through neighborhood searching are looking to find a better answer.

Step 7

If there are still unassigned customers, go to Step 1; otherwise, terminate the procedure and encode the current solution using the solution representation described in Sections 3.1.

3.3 Neighborhood

Explore the solution space do through various algorithms. Selecting proper method has a great affect on obtain better solution. So that we use 3 methods to get neighborhood solution from exiting solution(X). 2-OPT, 3-OPT and inverse 2-OPT are three approach that use in seeking neighborhood solution.

3.3.1 2-OPT

Selecting the i_{th} and the j_{th} numbers of X , and then exchanging the positions of these two numbers (Figure 3).



Figure 3. 2-OPT

3.3.2 3-OPT

Selecting the i_{th} and the j_{th} and k_{th} numbers of X , and then replace k_{th} number in i_{th} position and i_{th} number in j_{th} position and j_{th} number in k_{th} position. (Figure 4)



Figure 4. 3-OPT

3.3.3 Inverse 2-OPT

Selecting the i_{th} and the j_{th} numbers of X , and then reverse the number in the solution encoding between them. (Figure 5)



Figure 5. Inverse 2-opt

3.4 Heuristic SA Procedures

Simulated Annealing (SA) is motivated by an analogy to annealing in solids. The idea of SA comes from a paper published by Metropolis et al in 1953 (Metropolis, 1953). The algorithm in this paper simulated the cooling of material in a heat bath. This is a process known as annealing.

In 1982, Kirkpatrick et al (Kirkpatrick, 1983) took the idea of the Metropolis algorithm and applied it to optimization problems. The idea is to use simulated annealing to search for feasible solutions and converge to an optimal solution.

The SA uses seven parameters I_{iter} , T_0 , T_F , K , P , $N_{non-improving}$ and α . I_{iter} denotes the number of iterations the search proceeds at a particular temperature. T_0 represents the initial temperature, while T_F is the final temperature below which the SALRP procedure is terminated. K is the Boltzmann constant used in the probability function to determine whether to accept a worse solution or not. P is the unit penalty cost associated with the violation of depot capacity. This penalty term is set to be 0 in cases with uncapacitated depots. $N_{non-improving}$ is the maximum allowable number of consecutive temperature reductions during which the best objective function value is not improved. Finally, α is the coefficient controlling the cooling schedule.

In the beginning, the current temperature T is set to be the same as T_0 . Then an initial solution X is generated that described in Section 4.3. The current best solution X_{best} and the best objective function value obtained so far are set to be X and $obj(X, P)$, respectively. At each iteration the next solution Y is generated from $N(X)$ and its objective function value is evaluated. **Let Δ denote the difference between $obj(X, P)$ and $obj(Y, P)$, that is $\Delta = obj(Y, P) - obj(X, P)$.**

The probability of replacing X with Y , given that $\Delta > 0$, is $\exp(-\Delta/KT)$. This is accomplished by generating a random number $r \in [0,1]$ and replacing the solution X with Y if $r < \exp(-\Delta/KT)$. Meanwhile, if $\Delta < 0$, the probability of replacing X with Y is 1. X_{best} records the best solution found so far as the algorithm progresses. The current temperature T is decreased after running I_{iter} iterations since the previous decrease, according to the formula $T \rightarrow \alpha T$, where $0 < \alpha < 1$: After each temperature reduction, a local search procedure which sequentially performs swap and insertion is used to improve the current best solution.

The procedure is stop when the current temperature T is lower than T_F or the current best solution X_{best} is not improved in $N_{non-improving}$ consecutive temperature reductions. Following the termination of the SA procedure, the facility locations and vehicle routes can be derived from X_{best} . The proposed approach is summarized in Figure 6.

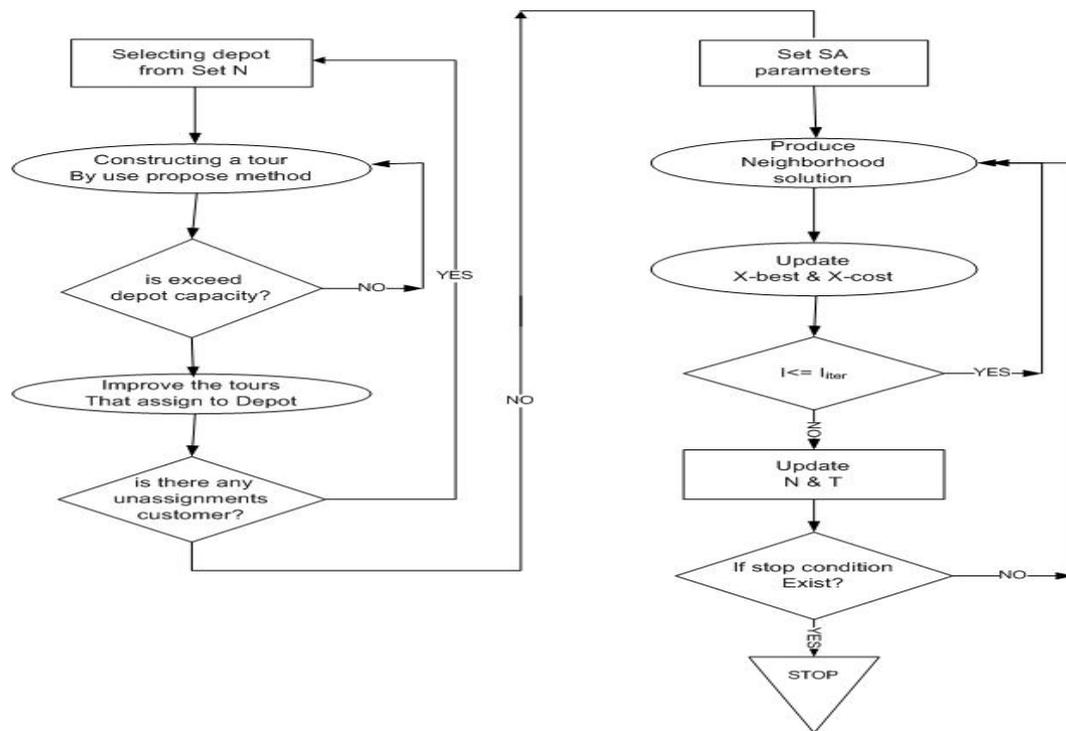


Figure 6. Proposed SA

4. Numerical Examples and Analysis

The proposed SA based meta-heuristic was implemented in MATLAB language and run on a PC with an Intel Core i5 CPU (2.27 GHz) and 2 GB memory. In this section, computational tests are carried out on 2 standard instances (CLRP) and solutions of Sahraeian method. Table 3 summarizes the results of the computational experiments. To consider efficiency of the resented heuristic method, its results are compared to Barreto et al. (2007) and Marinakis and Marinaki, (2008) (Note 1) and Sahraeian and nadizadeh (2009).

Table 3. Computational result

CLRP Instance	Vehicle capacity	BKS	Heuristic method	Gap %	Heuristic method	Sahraeian method	Gap %
Christofides69-50x5	160	582.7	565.6	-2.93	565.6	582.7	-2.93
Christofides69-75x10	140	886.3	896.9	1.3	896.9	886.3	1.2
Christofides69-100x10	200	889.4	891.5	0.24	891.5	889.4	0.24
Gaskell67-21x5	6000	432.7	424.9	-1.8	424.9	427.7	-0.65
Gaskell67-22x5	4500	587.9	585.1	-0.48	585.1	591.5	-1.08
Gaskell67-29x5	4500	512.1	512.1	0	512.1	522.4	-1.97
Gaskell67-32x5	8000	570.5	562.3	-1.45	562.3	567.2	-0.88
Gaskell67-32x5	11000	510.9	504.3	-1.31	504.3	504.3	0
Gaskell67-36x5	250	470.7	460.4	-2.19	460.4	469.2	-1.87
Per183-12x2	140	204	201.5	-1.22	201.5	205.3	-1.8
Per183-55x5	160	1121.1	1121.4	0	1121.4	1183.1	-5.2

In first column of Table 3, some examples with number of customers and candidate depots are presented by researches. Second column shows the capacity of the vehicle. The best solutions for these problems are in

column 3 and the results of proposed heuristic method are represented in forth column. Fifth column shows also the quality of the obtained solution as percent of deviation from best know solutions (BKS). It is calculated as $((\text{cost}_{\text{heuristic}} - \text{cost}_{\text{BKS}}) / \text{cost}_{\text{BKS}})$ that $\text{cost}_{\text{heuristic}}$ is the best solution cost obtains by proposed heuristic method and cost_{BKS} denotes the cost of the best known solution. The results of proposed heuristic method and Sahraeian method are represented in sixth and seventh column respectively. Eighth column shows also the quality of the obtained solution as percent of deviation from Sahraeian solutions. Column 9 shows the required average time to get the answer.

Parameter selection may influence the quality of the computational results, So that we consider below parameters for our new proposal method.

$$\alpha = 0.96, I_{\text{iter}} = 2000 * L, P = 1000, K = 1/10, T_0 = 35, T_f = 0.1, N_{\text{non-improvement}} = 150$$

As show in table 3 our method is better than answers of standard problems and Sahraeian method. The bold numbers in table 3 indicate the best deviation found. Our methods in front of standard problem improved 7 answers of standard problems. 1solution have not been changed and 3 problems had also worse answers. Our methods in front of Sahraeian method improved 8 answers of standard problems. 1solution have not been changed and 2 problems had also worse answers. On the whole, the quality and the time of answers of the method are effective and considerable.

5. Conclusion

In this paper, a heuristic method based on simulated annealing for Location-Routing problem is proposed. Our method present an heuristic with a special solution encoding scheme that integrates location and routing decisions in order to enlarge the search space so that better solutions can be found. The proposed SA algorithm was compared with two problem set and obtained solution represented that our method gets better solution. Our methods in front of standard problem improved 7 answers of standard problems. 1solution have not been changed and 3 problems had also worse answers. Our methods in front of Sahraeian method improved 8 answers of standard problems. 1solution have not been changed and 2 problems had also worse answers. On the whole, the quality and the time of answers of the method are effective and considerable.

Further two related research directions are as follows:

- (1) Develop a model and methods for the multi- product location-routing problem.
- (2) An integrated approach between SA and other metaheuristic algorithms may get better solution for location-routing problem.

References

- Duhamel, C., Lacomme, P., Prins, C., & Prodhon, C. (2008). A memetic approach for the capacitated location routing problem. In Proceedings of the EU/meeting 2008 workshop on metaheuristics for logistics and vehicle routing. University of Technology of Troyes, France.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671-680. <http://dx.doi.org/10.1126/science.220.4598.671>
- Laporte, G., & Norbert, Y. (1981). An exact algorithm for minimizing routing and operating costs in depot location. *European Journal of Operational Research*, 6, 224-226. [http://dx.doi.org/10.1016/0377-2217\(81\)90212-5](http://dx.doi.org/10.1016/0377-2217(81)90212-5)
- Laporte, G., Nobert, Y., & Arpin, D. (1986). An exact algorithm for solving a capacitated location routing problem. *Annals of Operations Research*, 6(9), 293-310. <http://dx.doi.org/10.1007/BF02023807>
- Laporte, G., Nobert, Y., & Pelletier, P. (1983). Hamiltonian location problems. *European Journal of Operational Research*, 12, 80-87. [http://dx.doi.org/10.1016/0377-2217\(83\)90182-0](http://dx.doi.org/10.1016/0377-2217(83)90182-0)
- Marinakos, Y., & Marinaki, M. (2008). A particle swarm optimization algorithm with path relinking for the location routing problem. *Journal of Mathematical modeling and algorithms*, 7, 59-78. <http://dx.doi.org/10.1007/s10852-007-9073-6>
- Melechovsky, J., Prins, C., & Calvo, R. W. (2005). A metaheuristic to solve a location-routing problem with non-linear costs. *Journal of Heuristics*, 11, 375-391.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculation by Fast Computing Machines. *J. of Chem. Phys.*, 21, 1087-109. <http://dx.doi.org/10.1063/1.1699114>

- Prins, C., Prodhon, C., Ruiz, A., Soriano, P., & Wolfler Calvo, R. (2007). Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation granular tabu search heuristic. *Transportation Science*, 41(4), 470–483. <http://dx.doi.org/10.1287/trsc.1060.0187>
- Sahraeian, R., & Nadizadeh, A. (2009). Using greedy clustering method to solve capacitated location-routing problem. 3rd International Conference on industrial Engineering and Industrial Management. Barcelona-Terrassa.
- Vincent F. Yu, Shih-Wei Lin, Wenyih Lee & Ching-Jung Ting. (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering*, 58, 288–299. <http://dx.doi.org/10.1016/j.cie.2009.10.007>
- Wu, T. H., Low, C., & Bai, J. W. (2002). Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, 29, 1393-1415. [http://dx.doi.org/10.1016/S0305-0548\(01\)00038-7](http://dx.doi.org/10.1016/S0305-0548(01)00038-7)

Note

Note 1. Instances are available in http://sweet.ua.pt/~iscf143/_private/SergioBarretoHomePage.htm