

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/304674287>

A Memetic Algorithm for the Capacitated Location–Routing Problem

Article in *International Journal of Advanced Computer Science and Applications* · July 2016

DOI: 10.14569/IJACSA.2016.070628

CITATIONS

2

READS

75

3 authors:



Laila Kechmane

12 PUBLICATIONS 7 CITATIONS

SEE PROFILE



Benayad Nsiri

Mohammed V University of Rabat

74 PUBLICATIONS 130 CITATIONS

SEE PROFILE



Azeddine Baalal

Faculté des Sciences Ain Chock - Casablanca

41 PUBLICATIONS 44 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



identification approach for supply chain systems [View project](#)



The discrete Laplacian on 3-simplicial complex [View project](#)

A Memetic Algorithm for the Capacitated Location-Routing Problem

Laila KECHMANE

Department of Mathematics and
Computing, MACS laboratory
Faculty of science, Hassan 2
University
Casablanca, Morocco

Benayad NSIRI

Department of Mathematics and
Computing, LIAD laboratory
Faculty of science, Hassan 2
University
Casablanca, Morocco

Azeddine BAALAL

Department of Mathematics and
Computing, MACS laboratory
Faculty of science, Hassan 2
University
Casablanca, Morocco

Abstract—In this paper, a hybrid genetic algorithm is proposed to solve a Capacitated Location-Routing Problem. The objective is to minimize the total cost of the distribution in a network composed of depots and customers, both depots and vehicles have limited capacities, each depot has a homogenous vehicle fleet and customers' demands are known and must be satisfied. Solving this problem involves making strategic decisions such as the location of depots, as well as tactical and operational decisions which include assigning customers to the opened depots and organization of the vehicle routing. To evaluate the performance of the proposed algorithm, its results are compared to those obtained by a greedy randomized adaptive search procedure, computational results shows that the algorithm gave good quality solutions.

Keywords—*hybrid genetic algorithm; capacitated location-routing problem; location; assigning; vehicle routing*

I. INTRODUCTION

Managing distribution is one of the most challenging problems for companies that aim to minimize the costs of their activities and meet the customers' needs in an environment where the competitiveness continues to increase.

Location problems and routing problems have long been treated separately, combining these two problems is the location and routing problem (LRP) whose objective is to optimize the costs of the distribution in a logistic network in taking strategic decisions that relate to the location of facilities, and allocation and organization of vehicle routing which are solved at the tactical and operational levels.

Salhi and Rand highlight the importance of tackling both problems simultaneously and the effect of ignoring routes when locating depots [1]. There are different formulations of the LRP, some consider depots and vehicles with limited capacities [2], while others, such as List and Mirchandani [3], offer formulations without capacity constraints. Wu et al. [4] have considered constraints on the vehicles and formulated a model for LRP with limited heterogeneous fleet, and Liu and Lee included inventory costs in LRP [5]. The reader is referred to the survey by Nagy and Salhi [6] for more LRP variants.

Exact methods have been proposed to solve the Capacitated Location-Routing Problem (CLRP), Laporte et al. [7] solved the problem through a branching method based on the relaxation of subtour elimination and the chain baring

constraints, Baldacci et al. [8] and Contardo et al. [9] found good quality solutions to the LRP using exact methods; Baldacci Proposed a branch and price algorithm and Contardo developed an algorithm based on cut-and-column generation for the CLRP. Prins et al. [10] solved the CLRP with a Greedy Randomised Adaptive Search Procedure (GRASP) followed by a path relinking algorithm and Duhamel et al. [11] proposed a GRASP hybridized with Evolutionary local search (GRASP X ELS) while Ting et al. [12] proposed a multiple ant colony optimization algorithm. Prodhon [13] presented a survey on the CLRP methods of resolution and other variants of the LRP. Our point of focus in this paper is on the evolutionary algorithms that are presented in the sequel.

Genetic algorithms have been proposed for the first time by J. Holland (1962), and then developed by D. Goldberg (1989), they are inspired by the natural evolution in genetics, where a population of individuals represents a set of solutions, an individual is represented by a chromosome and each chromosome contains genes.

Genetic algorithms have been successfully applied to the resolution of combinatorial optimization problems, several algorithms have been developed; Potvin [14] have presented a genetic algorithm for the Traveling Salesman Problem (TSP), Zhao [15] proposed a hybrid genetic algorithm to solve a TSP with pickup and delivery, while Prins [16] have presented an efficient genetic algorithm to solve the vehicle routing problem (VRP), Vidal et al. [17] have also used genetic algorithm to solve a multi-depot and periodic vehicle routing problem. For an overview of genetic algorithms, Reeves' book [18] is a good reference.

The first step of a genetic algorithm is to initialize the population and represent it in the form of chromosomes. The first generation is often randomly generated. The second step is to assess the individuals of a population in order to measure the goodness of each solution and select them according to their fitness in order to enable the best chromosomes to survive, then comes the stage of the crossover which involves crossing two chromosomes parents in order to obtain one or two new children called offspring. An offspring is better than the parents if it takes the better part of each. After the crossover, a mutation can be applied to the obtained children in order to prevent a premature convergence of the algorithm. In the incremental mode, only two parents breed and their children

are integrated in the current population, while in the generational mode, each iteration reproduces all the children who will be the next generation.

Genetic algorithms provide results of good qualities but hybridized with a local search, the results are better, the two methods complement each other; the first explores the different regions of the search space while the other exploits it intensively [19]. Local research is a powerful means to obtain quality solutions for optimization problems, it allows improving a solution by exploring its neighborhood, the latter is obtained by applying transformations to the solution, the best solution of this neighborhood is called a local minimum.

Genetic Algorithms hybridised to a local search are called Memetic algorithms. they have been introduced by Moscato in 1989 [19], they have led to excellent results for different problems, Krasnogor and Smith [20] have reviewed some applications of memetic algorithms to well known combinatorial optimization problems and Neri and Cotta have presented a literature review of these algorithms [21]. Various memetic algorithms have been developed for routing problems; Freisleben and Mertz [22] used it to solve the TSP, Cattaruzza et al. [23] for the multi-trip vehicle routing, Mendoza et al. [24] for the multi-compartment vehicle routing problem with stochastic demands, Prins [25] and Lima et al. [26] have proposed memetic algorithms to solve heterogeneous fleet vehicle routing problem. The most recent method has been proposed by Sörensen and Sevaux [27], it is a memetic algorithm with population management (MA/PM) that defines a measure of distance in order to diversify the chromosomes parents of the algorithm.

This paper aims to solve a capacitated location routing problem while minimizing the total cost which includes the depots opening cost, the vehicles cost of use and the routing cost using a memetic algorithm. The paper is organized as follows: Section II describes the problem, the algorithm and its components are presented in Section III, Section IV provides the parameters used and computational experiments and some concluding remarks are presented in Section V.

II. PROBLEM DESCRIPTION

In this paper a capacitated location routing problem is treated. Let $G = (V, E, C)$ be a weighted complete graph where $V = Dep \cup Nsc$, Dep is the set of potential depots to open and Nsc is the set of customers to serve, $E = \{(i, j) : i \in V, j \in V, i \neq j\}$ is the set of arcs linking the different nodes in the graph and $C = \{c_{ij} / i \in V, j \in V\}$ where c_{ij} is the cost of the trip via the arc (i, j) . Each depot i has a limited capacity Cap_i , a fixed cost of opening O_i and a homogeneous fleet F_i of vehicles v_{im} of a capacity cpv_{im} and a cost of use vc_{im} . There is no limit on the number of vehicles. Each customer j has a demand d_j which must be satisfied. A vehicle can serve multiple customers during his tour but a customer is served by only one vehicle. Each vehicle returns to

its depot of departure after the last visited customer. We assume that the overall capacity of the depots satisfies the demand of all customers and that the request of a customer does not exceed the capacity of a vehicle. The objective is to minimize the total cost which is the sum of the depots opening costs and the routing costs while responding to all the customers' requests.

III. A MEMETICALGORITHM FOR THE CLRP

In order to solve this problem, a memetic algorithm is proposed, this section includes the details of its different components.

A. Initial population

The initial population is obtained using a constructive algorithm based on the principle of the Nearest Neighbor Search (NNS), depots to be opened are randomly selected, which allows the exploration of the solutions space. In the sequel, Ct represents the solution cost, the variables x_i and y_{ijm} take respectively the value 1 if the depot i is open and if the customer j is served by the depot i via the vehicle v_{im} , 0 otherwise.

Fig. 1. presents the steps of the proposed algorithm, it starts by opening a depot selected in a random way, as long as its capacity allows, the closest customer to the last visited node which is not yet served is assigned to it, once the depot capacity no longer allows to provide customers, the approach is reiterated and another depot is opened until all customers are affected.

Note: During the initialization of the population, it is necessary to check that there are no clones, in case two chromosomes have equal costs, one of them is deleted and replaced by another one selected randomly.

B. Solution encoding

Solutions encoding is an important step that can influence the performance of genetic algorithms, in the proposed algorithm, the encoding method proposed by Prins [28] is adopted, each

Procedure: GenInitialSol

Input : Problem' data values

Output : Initial solution

$Ct \leftarrow 0$

repeat

Select randomly an available depot p

$x_p \leftarrow 1$

$Dep \leftarrow Dep - \{p\}$

$m \leftarrow 1$

$Ct \leftarrow Ct + O_p + vc_{pm}$

$j \leftarrow p$

repeat

$l \leftarrow \arg \min_{k \in Nsc} \{c_{jk} / cap_p > d_k \text{ and } d_k < cpv_{mp}\}$

```

 $y_{plm} \leftarrow 1$ 
 $Cap_p \leftarrow Cap_p - d_i$ 
 $Nsc \leftarrow Nsc - \{l\}$ 
 $cpv_{pm} \leftarrow cpv_{pm} - d_i$ 
 $Ct \leftarrow Ct + c_{pl}$ 
 $j \leftarrow l$ 
if ( $\forall k \in Nsc, d_k > cpv_{mp}$ ) then
     $m \leftarrow m + 1$ 
     $Ct \leftarrow Ct + c_{jp} + vc_{pm}$ 
end if
until ( $Cap_p < d_i$ )
until ( $Nsc = \emptyset$ )
return the initial solution

```

Fig. 1. The procedure to generate an initial solution

solution is represented as a chromosome, the latter indicates the status of the depots as well as the allocation of customers to opened depots. Each chromosome consists of two sequences, the first relates to the depots and the other to the customers, there is no trip delimiter. If a gene of the depots sequence is zero, this means that the depot is closed, otherwise it contains the index of the first customer assigned to it. The customers sequence is a concatenation of trips.

For each chromosome S , vectors ds and cs represent respectively the depots and the customers sequences. Fig. 2. represents an example of a chromosome, the depot 2 is closed because $ds(2) = 0$, the depot 1 is open, the set of customers affected to it is $C_1 \leftarrow \{1, 2, 3, 4\}$, this sequence begins with the first customer assigned to the depot and ends by the one just before the first customer assigned to another opened depot, depot 3 is also open and the set of customers affected to it is $C_3 \leftarrow \{5, 6, 7, 8, 9, 10\}$.

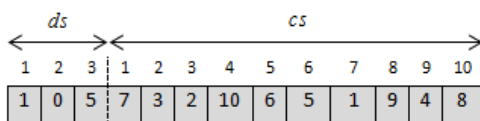


Fig. 2. Example of a chromosome

C. Selection and crossover

The first parent is selected using the binary tournament method on half of the best individuals of the population, the second also is chosen with this method but this time on the whole population except the first chosen parent. A single child is kept randomly, it gives better results than keeping the best one or both.

Crossover for the depots sequence acts as on a sequence of binary (one crossover point), a cut-off point is randomly chosen between 1 and $l_d - 1$ where l_d is the length of the depot sequence. The depot part of the child consists of the left part before the cut point of the first parent and the right part

after the cut point of the second parent. The classical order crossover (OX) is used to obtain the customers sequence, two cut points p and q are chosen randomly, such as $1 < p < q < l_c - 1$, where l_c is length of the customers sequence. The part between the two cut points of the first parent is copied in the same order and in the same position in the child chromosome, then the latter is supplemented by scanning the genes of the second parent from the position $q + 1$ up to l_c , then from the position 1 up to q and copying each element not yet in the child.

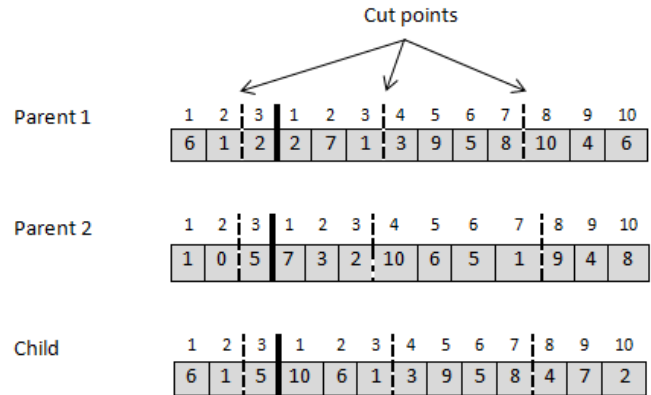


Fig. 3. Example of chromosomes crossover

D. Chromosome repair

After the crossover, it is necessary to check that the chromosome obtained is valid, we begin by verifying that all customers have been assigned, it is sufficient to check the existence of 1 in the depots sequence, it is also necessary to verify that the depots capacities have been respected, if this is not the case, customers should be removed one by one from such a the depot and assigned to the nearest one whose capacity allows, until all depots capacities are respected. If none of the opened depots allows it, a closed depot is opened. Fig. 4. describes the repair procedure steps. Variables C_i and Sd_i represent respectively the set of customers assigned to a depot i and the sum of customers assigned to this depot demands. The function $Close(j, E)$ returns the index of the nearest node to j in a set of nodes E .

Procedure: ChromRepair

Input : A chromosome S

Output : Repaired chromosome

Find $i / ds(i) = 1$, if it fails, find the first closed depots i' and $ds(i') \leftarrow 1$

for each $i \in Dep$ **do**

if ($ds(i) \neq 0$) **then**

for each $j \in C_i$ **do**

if ($Sd_i > Cap_i$) **then**

repeat

$j' \leftarrow \text{last customer affected to } i$

```

i' ← Close(i, Dep) /  $x_i = 1$  and  $Cap_{i'} > Sd_{i'} + d_j$ ,
Ci ← Ci + {j'}
Ci' ← Ci' + {j'}
if (i' = ∅) then
    Find i" /  $ds(i'') = 0$ 
    ds(i'') ← 1
    Ci ← Ci + {j'}
    Ci'' ← Ci'' + {j'}
end if
Until ( $Sd_i \leq Cap_i$ )
end if
end for
end if
end for
end if
return the repaired chromosome
    
```

Fig. 4. Chromosome repairing procedure

E. Solution cost

In order to obtain a solution cost, the procedure presented in Fig. 5. is proposed, it also allows deducing the vehicle routing organization; For each depot, as long as the capacity of the vehicle allows, the closest customer to the last one of the tour is added, once no more customers can be added, the vehicle returns to the depot, and another tour begins via another vehicle of the same depot fleet, it begins from the nearest customer of the depot which is not yet visited.

This procedure allows to obtain and to optimize the depots opening costs and the routing costs.

The function $Add(T_{im}, j)$ adds the node *j* at the end of the tour T_{im} via the vehicle v_{im} .

```

Procedure: SolCost
Input : A solution S
Output : Solution S cost
    
```

```

Ct ← 0
for each i ∈ Dep do
    if ( $ds(i) \neq 0$ ) then
        Ct ← 0
        j ← index of first customer affected to i
        m ← 1
        repeat
             $Add(T_{im}, j)$ 
            Ct ← Ct +  $vc_{im}$ 
             $cpv_{im} \leftarrow cpv_{im} - d_j$ 
            Ci ← Ci - {j}
            Ci' ← Ci' - {j}
        
```

```

Ct ← Ct +  $c_{ij}$ 
repeat
    j' ← Close(j, Ci')
    if ( $d_{j'} < cpv_{im}$ ) then
         $Add(T_{im}, j')$ 
         $cpv_{im} \leftarrow cpv_{im} - d_{j'}$ 
        Ct ← Ct +  $c_{jj'}$ 
        Ci ← Ci - {j'}
        Ci' ← Ci' - {j'}
    else
        Ci' ← Ci' - {j'}
    end if
until (Ci' = 0)
    Ct ← Ct +  $c_{ji}$ 
    j ← Close(i, Ci)
    m ← m + 1
until (Ci = 0)
end if
end for
Return the total cost of the solution, Ct
    
```

Fig. 5. The procedure to obtain a solution cost

F. Local search

The Local Search allows improving a solution by exploring its neighborhood, for this, four movements are used and presented in the sequel, each is illustrated by an example. The movements are applied to customers that can belong to the same tour or from different tours. Tours can belong to the same depot or to different ones, for each case the constraints are presented. Fig. 10. describes the steps of the procedure, the neighborhood of a solution is explored and the procedure returns the best solution and its cost.

In the sequel, *S* is a solution, Sd_i is the sum of demands that provides the depot *i* and Qt_{im} is the amount transported during a tour via the vehicle v_{im} .

RouteTrans(*S*, *i*, *i'*, *m*): a random tour T_{im} is transferred from its current depot *i* to the best opened one *i'*. Constraint (1) ensures that the capacity of the depot to which a tour is transferred is respected.

$$Cap_{i'} \geq Sd_{i'} + Qt_{im} \quad (1)$$

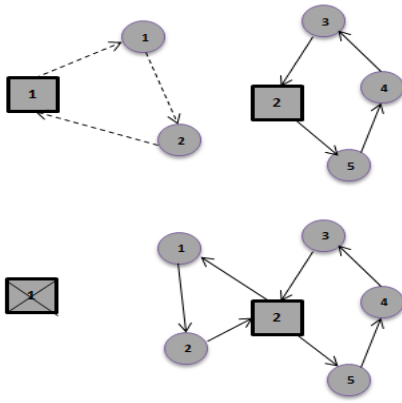


Fig. 6. Example of a RouteTrans operator

CustTrans (S, i, i', m, m', j, j') : a customer j , randomly selected, is transferred from its current position in a tour m to another position after a node j' in a tour m' that may be in the same route ($i = i'$ and $m = m'$) in another one from the same depot ($i = i'$ and $m \neq m'$) or from another route from another depot ($i \neq i'$ and $m \neq m'$). Constraints (2) and (3) ensure that both capacities of the vehicle and the depot, to which a customer is transferred, are respected.

$$cpv_{m'i'} \geq Qt_{i'm'} + d_j \quad (2)$$

$$Cap_{i'} \geq Sd_{i'} + d_j \quad (3)$$

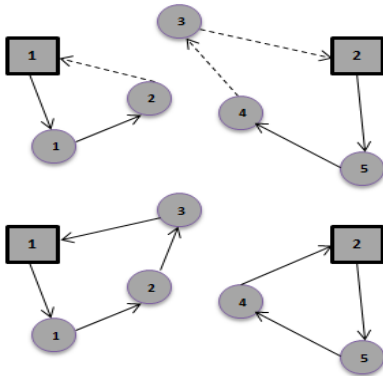


Fig. 7. Example of a CustTrans operator

CustSwap (S, i, i', m, m', j, j') : two customers randomly selected j and j' are exchanged, they may be in the same route ($m = m'$) or in different routes ($m \neq m'$) from the same depot ($i = i'$) or different depots ($i \neq i'$). Constraints (4)-(7) ensure that both capacities of the vehicles and the depots, to which customers are transferred are respected.

$$cpv_{i'm'} \geq Qt_{i'm'} + d_j - d_{j'} \quad (4)$$

$$cpv_{im} \geq Qt_{im} + d_{j'} - d_j \quad (5)$$

$$Cap_{i'} \geq Sd_{i'} + d_j - d_{j'} \quad (6)$$

$$Cap_i \geq Sd_i + d_{j'} - d_j \quad (7)$$

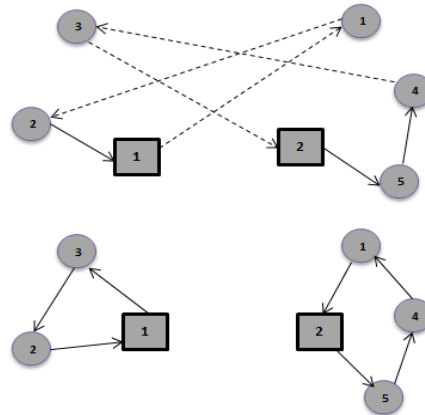


Fig. 8. Example of a CustSwap operator

Opt (S, i, m, m', j, j') : the edges incoming to nodes j and j' are deleted and two new edges are created, tours begin from the same depot i and edges may be in the same tour ($m = m'$) or in different tours ($m \neq m'$). Constraints (8)-(11) ensure that both capacities of the new tours' vehicles and their departure depots are respected.

$$cpv_{im} \geq Qt_{im} + d_{j'} - d_j \quad (8)$$

$$cpv_{i'm'} \geq Qt_{i'm'} + d_j - d_{j'} \quad (9)$$

$$Cap_i \geq Sd_i + d_{j'} - d_j \quad (10)$$

$$Cap_{i'} \geq Sd_{i'} + d_j - d_{j'} \quad (11)$$

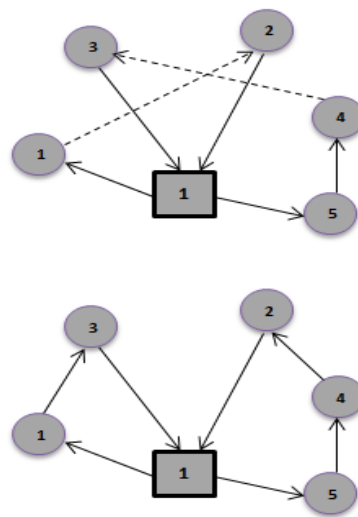


Fig. 9. Example of an Opt operator

Procedure: Localsearch
Input : A solution S
Output : S_{best}

$S_{best} \leftarrow S$; $S_{min} \leftarrow S$
 $N_{RT} \leftarrow \emptyset$; $N_{CT} \leftarrow \emptyset$; $N_{CS} \leftarrow \emptyset$; $N_{OPT} \leftarrow \emptyset$
 $N_{RT} \leftarrow S$ neighborhood generated using RouteTrans operator
 $N_{CT} \leftarrow S$ neighborhood generated using CustTrans operator
 $N_{CS} \leftarrow S$ neighborhood generated using CustSwap operator
 $N_{OPT} \leftarrow S$ neighborhood generated using Opt operator
 $S_{min} \leftarrow \arg \min \{SolCost(S) / S \in N_{RT} \cup N_{CT} \cup N_{CS} \cup N_{OPT}\}$
Repair S_{min} using ChromRepair procedure
if $SolCost(S_{min}) < SolCost(S_{best})$ then
 $S_{best} \leftarrow S_{min}$
end if
Return S_{best} and $SolCost(S_{best})$

Fig. 10. The local search procedure

G. The memetic algorithm

The memetic algorithm is obtained by gathering the components previously presented, it starts with generating an initial population using GenInitialSol procedure, a maximum number of generations $\max gen$ is fixed, as long as the latter is not reached the selection and the crossover are repeated, each child obtained is repaired with ChromRepair procedure, if it does not already belong to the population it is improved using LocalSearch procedure with a probability p_{ls} and replaces the worst element of the population, once $\max gen$ is reached, the procedure returns the best solution of the last obtained population. The Local Search is applied only once to each offspring, which allows exploring its neighborhood, applying it several times would take a lot of time and would lead to the local optimum.

Procedure: MemeticAlgorithm
Input : Problem data
Output : Best solution P_{best}

$t \leftarrow 0$
Initialize population $P(t)$ using GenInitialSol procedure
while ($t < \max gen$) **do**
 Select parents p_1 and p_2 using binary tournament

Cross p_1 and p_2 to obtain child c
Repair c using ChromRepair procedure
If $c \in P(t)$ **then**
 Repeat selection and crossover
else
 Apply LocalSearch procedure to c with a probability p_{ls}
 to obtain c'
 if $c' \notin P(t)$ **then**
 remove worst solution from $P(t)$ and add c' to $P(t)$
 end if
 $t \leftarrow t + 1$
end while
Return P_{best}

Fig. 11. General structure of the memetic algorithm

IV. COMPUTATIONAL EXPERIMENTS

In order to investigate the performance of the algorithm, test instances presented by Prodhon [28] for the CLRP are used, which consists of capacitated depots and routes, the number of potential depots is $N_d \in \{5, 10\}$, the number of customers to serve is $N_c \in \{20, 50, 100, 200\}$, and vehicle capacity is in $\{70, 150\}$. The depots opening costs vary from one depot to another, yet, the cost of use of a vehicle remains the same for all. After preliminary testing, p_{ls} is fixed to 0.6, $\max gen$ to $(N_d + N_c) * 20$ and the population size is $N_d + N_c$. Procedures are implemented in C language on an Acer Aspire ONE D255 1.00 GHz machine, running Windows 7 Starter Edition.

Table. 1. Provides a comparison between the results obtained by the proposed memetic algorithm and those obtained by the GRASP proposed by Prodhon, Prodhon's set consists of 30 instances, results are given on average by instances size. For each case, the number of depots to open, the number of tours and the solution cost are presented. The gap between the two algorithms is obtained by calculating $[(MA \text{ cost} - GRASP \text{ cost}) / GRASP \text{ cost}] * 100$. As shown in the table, good quality solutions are obtained by MA, it presents a gain in term of cost which is in average of 1.08%. The best improvements have been obtained for the large sizes where the gain reaches 2.73%. Both MA and GRASP returns the same number of depots to be opened, MA uses one less vehicle on certain instances from 100 customers, costs are reduced thanks to the opened depots location and not to their number, MA provides access to diverse solutions of good quality, which allows obtaining lower cost solutions.

TABLE I. COMPUTATIONAL RESULTS OF THE MEMETIC ALGORITHM

Number of potential depots	Number of customers	GRASP			MA			Gap
		Number of opened depots	Number of tours	cost	Number of opened depots	Number of tours	cost	
5	20	2.5	4	45144	2.5	4	45301	0.35
5	50	2.62	9	74701	2.62	9	74685	0.02
5	100	2.33	16	202210	2.33	15.83	201303	0.45
10	100	3.33	18.33	257048	3.33	17.66	250040	2.73
10	200	3	35	447657	3	34.5	436293	2.54
Average		2.76	16.47	205352	2.76	16.2	201524.4	1.08

V. CONCLUSION

This paper treats a location-routing problem where both potential depots and vehicles have limited capacities. A memetic algorithm is developed to solve the problem which objectives are to minimize the total cost of distribution and meet the customers' needs. The algorithm is tested on instances of the literature and compared to a GRASP. The proposed algorithm provides obtaining good quality solutions, it may be used to obtain a good initial solution to exact methods.

REFERENCES

- [1] S. Salhi, and G. K. Rrand, "The effect of ignoring routes when locating depots," in *European Journal of Operational Research*, vol. 39, no. 2, pp. 150–156, 1989.
- [2] J. H. Lee, I. K. Moon, and J. H. Park, "Multi-level supply chain network design with routing," in *International Journal of Production Research*, vol. 48, no. 13, pp. 3957–3976, 2010.
- [3] G. List, and P. Mirchandani, "An integrated network/planar multiobjective model for routing and siting for hazardous materials and wastes," in *Transportation Science*, vol. 25, no. 2, pp. 146–156, 1991.
- [4] T. H. Wu, C. Low, and J. W. Bai, "Heuristic solutions to multi-depot location-routing problems," in *Computers and Operations Research*, vol. 29, pp.1393–1415, 2002.
- [5] S. C. Liu, and S. B. Lee, "A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into considerations," in *International Journal of Advanced Manufacturing Technology*, vol. 22, pp. 941–950, 2003.
- [6] G. Nagy, and S. Salhi, "Location-routing: Issues, models and methods," in *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.
- [7] G. Laporte, Y. Norbert, and D. Arpin, "An exact algorithm for solving a capacitated location-routing problem," in *Annals of Operations Research*, vol. 6, pp. 293–310, 1986.
- [8] R. Baldacci, A. Mingozzi, and R.W.Calvo, "An exact method for the capacitated location-routing problem," *Operations Research*, vol. 59, no. 5, pp. 1284–1296, 2011.
- [9] C. Contardo, J. F. Cordeau, and B. Gendron, "An exact algorithm based on cut-and-column generation for the capacitated location-routing problem," in *INFORMS Journal on Computing*, vol. 26, no. 1, pp. 88–102, 2014.
- [10] C. Prins, C. Prodhon, and R. Calvo, "Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking," in *4OR: A Quarterly Journal of Operations Research*, vol. 4, pp. 221–238, 2006.
- [11] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon, "A GRASP \times ELS approach for the capacitated location-routing problem," *Computers and Operations Research*, vol. 37, no. 11, pp. 1912–1923, 2010.
- [12] C. J. Ting, and C. H. Chen, "A multiple ant colony optimization algorithm for the capacitated location-routing problem," in *International Journal of Production Economics*, vol. 141, no. 1, pp. 34–44, 2013.
- [13] C. Prins, and C. Prodhon, "A survey of recent research on location-routing problems," in *European Journal of Operational Research*, vol. 238, no. 1, pp. 1–17, 2014.
- [14] J. Y. Potvin, "Genetic Algorithms for Traveling Salesman Problem," in *Annals of Operations Research*, vol. 63, no. 3, pp 337–370, 1996.
- [15] F. G. Zhao, J. S. Sun, S. J. Li, and W. M. Liu, "A hybrid genetic algorithm for the traveling salesman problem with pickup and delivery," in *International Journal of Automation and Computing*. vol. 6, no. 1, pp. 97–102, 2009.
- [16] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," in *Computer and Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.
- [17] T. Vidal, T. G. Crainic, M. Gendreau, and W. Rei, "A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems," in *Operations Research*, vol. 60, no. 3, pp. 611–624, 2012.
- [18] C.R. Reeves, "Genetic algorithms," in *handbook of metaheuristics*, International series in operations research and management science, 2nd ed., Springer, 2010, pp. 109–140.
- [19] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," in *Technical Report C3P 826*, Cal-tech Concurrent Computation Program, 1989.
- [20] N. Krasnogor, and J. Smith, "A tutorial for competent memetic algorithms : model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005.
- [21] F. Neri, and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," in *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.
- [22] M. Freisleben, and P. Mertz, "Memetic Algorithms for the Traveling Salesman Problem," in *Complex Systems*, vol. 13, pp. 297–345, 2001.
- [23] D. Cattaruzza, N. Absi, and D. Feillet, "A memetic algorithm for the multi trip vehicle routing problem," in *European Journal of Operational Research*, vol. 236, issue. 3, pp 833–848, 2014.
- [24] J. Mendoza, B. Castanier, C. Gueret, A. Medaglia, and N. Velasco, "A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands," in *Computers and Operations Research*, vol. 37, issue. 11, pp. 1886–1898, 2010.
- [25] C. Prins, "Two memetic algorithms for heterogeneous fleet vehicle routing problems," in *Engineering Applications of Artificial Intelligence*, vol. 22, pp. 916–928, 2009.

- [26] C. Lima, M. Goldberg, and E. Goldberg, "A memetic algorithm for heterogeneous fleet vehicle routing problem," in *Electronic Notes in Mathematics*, vol. 18, pp. 171–176, 2004.
- [27] K. Sörensen, and M. Sevaux, "MA/PM : Memetic algorithms with population management," in *Computers and Operations Research*, vol. 33, pp. 1214–1225, 2006.
- [28] C. Prins, C. Prodhon, and R. Calvo, "A memetic algorithm with population management (MA/PM) for the capacitated location-routing problem," in *Evolutionary computation in combinatorial optimization*, J. Gottlieb, G. Raidl, Eds. Berlin/Heidelberg: Springer, 2006, pp. 183–94.