A Memetic Approach for the Capacitated Location Routing Problem

Christophe Duhamel* Philippe Lacomme* Christian Prins†
Caroline Prodhon†

*Laboratoire d'Informatique (LIMOS, UMR CNRS 6158),
Campus des Cézeaux, 63177 Aubière CEDEX
{placomme,christophe.duhamel}@sp.isima.fr

†Institut Charles Delaunay (FRE CNRS 2848), Université de Technologie de Troyes
BP 2060, 10010 Troyes Cedex, France
{christian.prins,caroline.prodhon}@utt.fr

Abstract

This paper addresses distribution network design problems that involves depot location, fleet assignment and routing decisions. The distribution networks under investigation are characterized by several depots, a capacitated homogeneous vehicle fleet and a set of customers nodes to be serviced with demands. The objective is to assign the serviced nodes to depots and to design the vehicle routes. The optimal solution minimizes both the depot cost and the total route distance in such a way that the total customer demand assigned to one depot is upper bounded by the depot capacity. A memetic algorithm is designed including a heuristic for initial generation of chromosomes, a powerful local search scheme and an efficient crossover procedure. The evaluation is made by the split procedure that takes into account the vehicle capacity, the number of vehicles, the depot capacity and the total cost. The framework is benchmarked on classical instances. The results prove that the method competes for small and medium scale instances with the best existing methods. New best solutions are even obtained.
Keywords: VRP, hub location, genetic algorithm.

Introduction

In supply chain management, one of the most challenging problems consists in a proper coordination of depot location and vehicle routing decisions. Strategies which solve consecutively the assignment of customers to hubs and the routing problem lead to suboptimal solutions [13]. The Location-Routing Problem (LRP) integrates these two decision levels with the objective of solving simultaneously both routing and location problems. Min *et al.* [7] provide a classification of the variants of the LRP according to considerations on: depot capacity, homogeneous or heterogeneous fleet of vehicles, fixed cost of vehicles. Mathematical formulations have been introduced with two or three indexes [4]. Exact solution schemes have been investigated in

[6, 5, 2] but are limited to medium scale instances or on basic uncapacitated instances. Numerous heuristic and meta-heuristic approaches have been introduced, including for instance [14, 15, 1]. However, problems including capacities constraints on both depots and routes (*general LRP*) has received less attention except the last years. We can quote Wu *et al.* [16] who divided this problem into two subproblems: a Location-Allocation Problem (LAP), and a Vehicle Routing Problem (VRP), solved in a sequential and iterative manner by a Simulated Annealing (SA) algorithm with a tabu list to avoid cycling. Barreto [2] developed a family of three-phase heuristics based on clustering techniques. Prins et al. have also developed algorithms on the general LRP. The first one is a GRASP (Greedy Randomized Adaptive Search Procedure) complemented by a post-optimization based on a path relinking algorithm [10]. The second one is a Memetic Algorithm with Population Management (MA|PM) [9]. The last one is a cooperative metaheuristic called LRGTS which alternates between a depot location phase and a routing phase, sharing some information [11].

The addressed problem is defined on a complete, weighted and directed network with a capacitated homogeneous fleet of vehicles. The following notations are used:

V	set of nodes including serviced nodes J and depot nodes I
J	set of customer nodes to service $J = \{1, 2, \dots, n\}$
I	set of depot nodes $I = \{1, 2, \dots, m\}$
O_i	opening cost induced by assignment of one customer to the depot i
W_i	depot i capacity
d_j	demand of customer node j
K	number of available vehicles
Q	vehicle capacity
F	fixed cost of a vehicle
c_{ij}	traveling cost from node i to j

A solution of the problem consists in defining which depots must be opened, assigning each serviced node to one and only one depot and routing the vehicle for nodes. The following constraints must be taken into account: (i) each serviced node is assigned to one depot;(ii) the total demand of serviced nodes assigned to one depot is less or equal to the depot capacity; (iii) each route starts and ends at the same depot; (iv) the total demand of serviced nodes assigned to one vehicle is less or equal to the vehicle capacity. Let us note $y_i = 1$ iif depot i is opened, $f_{ij} = 1$ iif customer j is assigned to depot i and $x_{ijk} = 1$ iif the arc $[i; j]$ is used in the route performed by vehicle k. The objective function Z is composed of depot opening cost $\sum_{i \in I} O_i Y_i$, vehicle fixed cost $\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} F x_{ijk}$ and traveling cost $\sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk}$.

The proposed solution method is a memetic algorithm (genetic algorithm hybridized with a local search procedure) able to deal with each level of decisions at the same time. It differs from the MAPM [9] by the way of encoding a chromosome. In [9], a chromosome is composed of two parts, one dealing with the depots status (open/close) and the assignment of customers to the open depots (depot sequence) and one with the routing (customer sequence). Here, the idea is to strengthen the evaluation of the fitness by encoding the chromosome with only a customer sequence, without trip or depot delimiters. Then, the fitness is calculated thanks to a Split procedure taking into account all the decisions with respect to the fleet vehicle capacity, the number of vehicles, the depot capacity and the total cost. This evaluation is explained in Section 1. The genetic scheme is complemented by local searches. The framework of the method is summarized in Section 2. The numerical experiments are in Section 3 and the paper ends by a conclusion and some perspectives.

1 A Split procedure for a permutation customer list evaluation

Successful domain applications of Split include the memetic algorithm of Lacomme *et al.* [3] for the CARP and the genetic algorithm of Prins [8] for the VRP. This successful approach tackles a permutation of customers fully defined by a permutation customer list $\lambda = (\lambda_1, \ldots, \lambda_n)$ where λ_i is the i^{th} customer to serve, without any consideration of vehicle and depot. At any time, a permutation λ can be converted into an optimal LRP solution (subject to the order imposed by λ), thanks to a special splitting procedure. This design choice provides a natural topological order of nodes and avoids repair procedures and enables the use of classical local search scheme. The split procedure works on an auxiliary graph $H = (X; A; Z)$. H is a set of $n + 1$ nodes indexed from 0 to n. An arc from nodes $i - 1$ to j represents a trip servicing nodes λ_i to λ_j. The weight z_{ij}^k of (i, j) is equal to the trip cost if depot k is used. A trip (i, j) servicing customers λ_i to λ_j is: vehicle capacity-feasible if $\sum_{r=i+1}^{j} d_{\lambda_r} \leq Q$ (C_1) and depot-feasible if $\sum_{r=i+1}^{j} d_{\lambda_r} \leq W_k^i$ (C_2).

The weight is $z_{ij}^k = O_k y_k + F + c_{\lambda_k \lambda_i} + c_{\lambda_{i+1} \lambda_k} + \sum_{r=1}^{r=j-1} c_{\lambda_r \lambda_{r+1}}$ with $y_k = 1$ if $W_k^i = W_k$ (no customer has been assigned to the depot) and $y_k = 0$ if at least one trip has been previously assigned to depot k. A node label $L_i^p = (K_i, W_1^i, \ldots, W_m^i, z_i^p, k, j)$ is the p^{th} label assigned to the node i and it is composed of: K_i (the number of available vehicles), W_d^i (the remaining capacity of the depot d), z_i^p (the objective function value to service customers λ_1 to λ_i), (k, j) (the father label of L_i^p we mean L_j^k the k^{th} label of node j).

The initial label of node 0 is $L_0 = (K_0, W_1, \ldots, W_m, 0, -1, -1)$ which represents a solution where $K_0 = K$ vehicles are available, and all the initial capacity of the depot is available ($W_1^0 = W_1, \ldots, W_m^0 = W_m$). The cost of the initial label is set to 0 ($z_0 = 0$). The pair (-1,-1) means this initial label has no predecessor in the graph. Each label $L_i^p = (K_i, W_1^i, \ldots, W_m^i, z_i^p, k, j)$ generates $L_j^r = (K_j, W_1^j, \ldots, W_m^j, z_i^r, i, p)$ using arc (i, j) and the weight $z_{ij}^k = O_k y_k + F + c_{\lambda_k \lambda_i} + c_{\lambda_{i+1} \lambda_k} + \sum_{r=1}^{r=j-1} c_{\lambda_r \lambda_{r+1}}$ (satisfying condition (C1) and (C2)) with: (i) $K_j = K_i - 1$; (ii) $W_k^j = W_k^i - \sum_{r=i+1}^{j} d_{\lambda_r}$; (iii) $z_j = z_i + z_{ij}^k$ with $y_k = 1$ if $W_k^i = W_k^0$ and $y_k = 0$ otherwise.

A label L_i^p can generate m new labels for node j provided that condition (2) holds. Note that j varies from $i + 1$ to n_i^- where $n_i^- = \arg\max(j | \sum_{r=i+1}^{j} d_{\lambda_r} \leq Q)$. n_i^- is the rank of the last customer which can be assigned to the trip starting with λ_i without exceeding the vehicle capacity. Trying to avoid excessive label generation, dominated feasible trips are discarded thanks to the following domination rules.

A label L_i^p **dominates** L_i^q if one of the following conditions holds:

$K_i < K_j$ and $\forall q = 1, \ldots, m \quad W_q^i \leq W_q^j$ and $z_i \leq z_j$

or $\exists q \in 1, \ldots, m \quad W_q^i < W_q^j$ and $K_i \leq K_j$ and $z_i \leq z_j$ and $\forall v = 1, \ldots, m \quad v \neq q \quad W_v^i \leq W_v^j$

or $z_i < z_j$ and $K_i \leq K_j$ and $\forall q = 1, \ldots, m \quad W_q^i \leq W_q^j$

An optimal splitting of a permutation $\lambda = (\lambda_1, \ldots, \lambda_n)$ can be obtained by storing only non-dominated labels for each node of the graph $H = (X, A, Z)$. An optimal LRP solution for $\lambda = (\lambda_1, \ldots, \lambda_n)$ corresponds to a min-cost path from 0 to n in H. This evaluation is reasonably fast thanks to the dominance rules presented above.

2 Framework

The framework is based on an incremental memetic method: a genetic algorithm (for generation of permutation customer lists) coupled with a powerful local search procedure. Such framework as been proved to be efficient in numerous routing problem including the CARP and the VRP [8]. The problem is modelled as a fully directed graph in which each arc represents a shortest path between two nodes. The Split procedure permits to assign one solution to each permutation and to define both customers assignment to depot and routes for vehicles. Three classical heuristics denoted H1, H2, H3 and a saving heuristic [10, 9] are used for the population initialization and during the restarts of the memetic algorithm.

3 Numerical experiments

All procedures are implemented under Borland C++ 6.0 package and experiments were carried out on a 2.4 GHz computer under Windows XP with 2 Gb of memory. The benchmark is composed of instances based on Prins et al.'s instances [10], Barreto's instances [2] and Tuzun and Burke's instances. Each instance is solved five times: table 1, table 2 and table 3 report the best run (Cost) for each instance and compare it to the lower bound (LB) from [12] and [2] or Tuzun and Burke solutions (Tuzun) from [15] (Gap/LB ou Gap/Tuzun, given in percentage). The framework (Proposal) outperforms all the previous methods on Barreto's instances, even closing the gap with the lower bound on one instance (Gaskell67-32x5). The proposed algorithm also provides better results than the GRASP [10] for all the instances and competes with LRGTS [11] on Tuzun's instances. Note that GRASP [10], MAPM [9] and LRGTS [11] results are obtained by only one run since the methods are very robust and does not required several experiments to provide a fair comparative study. Costs in boldface refer to the best solution.

4 Concluding remarks

A memetic algorithm is proposed for the LRP. It provides stated of the art solutions, however it is time consuming. The next work is to reduce the computational time. Furthermore, this research is a step toward resolution of more realistic problems which could include: (i) Heterogeneous fleet of vehicles;(ii) Prohibited turns in graph; (iii) Time-windows on customers services.

		Proposal			GRASP			MAPM			LRGTS		
	LB	Cost	CPU	gap/LB	Cost	CPU	gap/LB	Cost	CPU	gap/LB	Cost	CPU	gap/LB
20-5-1a	54793.00	**54793**	0.0	0.00	55021	0.2	0.42	**54793**	0.3	0.00	55131	0.4	0.62
20-5-1b	39104.00	**39104**	0.0	0.00	**39104**	0.2	0.00	**39104**	0.3	0.00	**39104**	0.2	0.00
20-5-2a	48908.00	**48908**	0.0	0.00	**48908**	0.1	0.00	**48908**	0.4	0.00	**48908**	0.5	0.00
20-5-2b	37542.00	**37542**	0.0	0.00	**37542**	0.2	0.00	**37542**	0.3	0.00	**37542**	0.1	0.00
50-5-1	84750.65	**90111**	6.0	6.32	90632	1.8	6.94	90160	2.6	6.38	90160	0.3	6.38
50-5-1b	59574.89	63469	58.0	6.54	64761	1.8	8.71	**63242**	3.2	6.16	63256	1.0	6.18
50-5-2	82057.13	88709	35.0	8.11	88786	2.4	8.20	**88298**	3.4	7.61	88715	1.8	8.11
50-5-2b	63841.35	**67353**	65.0	5.50	68042	2.5	6.58	67893	2.9	6.35	67698	1.8	6.04
50-5-2bis	82356.61	84409	28.0	2.49	**84055**	1.7	2.06	**84055**	3.2	2.06	84181	2.0	2.22
50-5-2bbis	51085.29	51902	27.0	1.60	52059	2.6	1.91	**51822**	4.2	1.44	51992	0.9	1.77
50-5-3	82703.76	**86203**	39.0	4.23	87380	2.3	5.65	**86203**	3.1	4.23	**86203**	0.3	4.23
50-5-3b	59473.83	62763	17.0	5.53	61890	2.0	4.06	**61830**	4.9	3.96	**61830**	0.5	3.96
100-5-1	272082.37	281564	220.0	3.48	279437	27.6	2.70	281944	26.3	3.62	**277935**	8.7	2.15
100-5-1b	207037.38	219056	226.0	5.81	216159	23.2	4.41	216656	34.5	4.65	**214885**	8.3	3.79
100-5-2	186916.59	197156	126.0	5.48	199520	17.4	6.74	**195568**	35.8	4.63	196545	2.3	5.15
100-5-2b	153827.05	159615	342.0	3.76	159550	22.4	3.72	**157325**	36.4	2.27	157792	3.3	2.58
100-5-3	194202.03	203723	188.0	4.90	203999	21.6	5.04	**201749**	28.7	3.89	201952	2.4	3.99
100-5-3b	149985.58	154404	291.0	2.95	154596	20.3	3.07	**153322**	33.3	2.22	154709	2.9	3.15
100-10-1	258242.64	325357	401.0	25.99	323171	37.4	25.14	316575	24.7	22.59	**291887**	14.1	13.03
100-10-1b	218825.96	274379	655.0	25.39	271477	29.5	24.06	270251	36.0	23.50	**235532**	14.0	7.63
100-10-2	226904.99	248331	306.0	9.44	254087	39.1	11.98	**245123**	24.6	8.03	246708	14.4	8.73
100-10-2b	194627.72	208508	801.0	7.13	206555	29.8	6.13	205052	31.6	5.36	**204435**	10.1	5.04
100-10-3	222353.23	264547	176.0	18.98	270826	35.4	21.80	**253669**	29.0	14.08	258656	13.3	16.33
100-10-3b	189308.50	211925	359.0	11.95	216173	39.8	14.19	**204815**	36.5	8.19	205883	10.8	8.76
Avg				**6.9**			**7.2**			**5.9**			**5.0**

Table 1: Solutions on Prins *et al*'s instances

References

[1] M. Albareda-Sambola, J.A Díaz, and E. Fernández. A compact model and tight bounds for a combined location-routing problem. *Computers and Operations Research*, 32:407–428, 2005.

[2] S.S. Barreto. *Análise e Modelização de Problemas de localização-distribuição [Analysis and modelling of location-routing problems]*. PhD thesis, University of Aveiro, campus universitário de Santiago, 3810-193 Aveiro, Portugal, October 2004. [In Portuguese].

[3] P. Lacomme, C. Prins, and W. Ramdane-Chérif. Competitive memetic algorithms for arc

		Proposal			GRASP			MAPM			LRGTS		
	Tuzun	Cost	CPU	gap/Tuzun	Cost	CPU	gap/Tuzun	Cost	CPU	gap/Tuzun	Cost	CPU	gap/Tuzun
111112	1556.64	**1487.35**	628.0	-4.45	1525.25	32.4	-2.02	1493.92	31.5	-4.03	1490.82	3.3	-4.23
111122	1531.88	1483.48	922.0	-3.16	1526.90	40.7	-0.32	**1471.36**	35.6	-3.95	1471.76	6.5	-3.92
111212	1443.43	1444.70	507.0	0.09	1423.54	27.6	-1.38	1418.83	36.2	-1.70	**1412.04**	4.2	-2.17
111222	1511.39	1466.92	1194.0	-2.94	1482.29	36.2	-1.93	1492.46	**36.4**	-1.25	**1443.06**	7.4	-4.52
112112	1231.11	1185.45	333.0	-3.71	1200.24	27.7	-2.51	**1173.22**	31.9	-4.70	1187.63	6.9	-3.53
112122	1132.02	1115.49	1381.0	-1.46	1123.64	34.3	-0.74	**1115.37**	42.7	-1.47	1115.95	6.8	-1.42
112212	825.12	807.85	557.0	-2.09	814.00	22.5	-1.35	**793.97**	38.0	-3.78	813.28	5.2	-1.43
112222	740.54	737.19	959.0	-0.45	747.84	37.3	0.99	**730.51**	49.3	-1.35	742.96	5.9	0.33
113112	1316.98	**1251.01**	877.0	-5.01	1273.10	21.5	-3.33	1262.32	36.8	-4.15	1267.93	4.3	-3.72
113122	1274.50	1260.10	1142.0	-1.13	1272.94	36.0	-0.12	**1251.32**	47.7	-1.82	1256.12	6.3	-1.44
113212	920.75	909.98	465.0	-1.17	912.19	20.3	-0.93	**903.82**	35.1	-1.84	913.06	4.0	-0.84
113222	1042.21	1036.86	1009.0	-0.51	1022.51	38.4	-1.89	**1022.93**	62.6	-1.85	1025.51	4.9	-1.60
Avg				**-2.2**			**-1.3**			**-2.7**			**-2.4**

Table 2: Solutions on Tuzun and Burke's instances

routing problems. *Annals of Operations Research*, 131:159–185, 2004.

[4] G. Laporte. Location-routing problems. In B.L. Golden and A.A. Assad, editors, *Vehicle Routing: Methods and Studies*, pages 163–196, Amsterdam, 1988. North Holland.

[5] G. Laporte, F. Louveaux, and H. Mercure. Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research*, 39:71–78, 1989.

[6] G. Laporte, Y. Norbert, and P. Pelletier. Hamiltonian location problems. *European Journal of Operational Research*, 12:82–89, 1983.

		Proposal			GRASP			MAPM			LRGTS		
	LB	Cost	CPU	gap/LB	Cost	CPU	gap/LB	Cost	CPU	gap/LB	Cost	CPU	gap/LB
Christofides69-50x5	551.1	584.8	80.0	6.11	599.1	2.3	8.71	**565.6**	3.8	2.63	586.4	2.4	6.41
Christofides69-75x10	791.4	**851.8**	207.0	7.63	861.6	9.8	8.87	866.1	9.4	9.43	863.5	10.1	9.11
Christofides69-100x10	818.1	**842.4**	408.0	2.97	861.6	25.5	5.31	850.1	44.5	3.91	842.9	28.2	3.02
Daskin95-88x8	347.0	355.9	582.0	2.55	356.9	17.3	2.83	**355.8**	34.2	2.52	368.7	17.5	6.25
Gaskell67-21x5	424.9	**424.9**	0.0	0.00	429.6	0.2	1.10	**424.9**	0.3	0.00	**424.9**	0.2	0.00
Gaskell67-22x5	585.1	**585.1**	0.0	0.00	**585.1**	0.2	0.00	611.8	0.3	4.56	587.4	0.2	0.39
Gaskell67-29x5	512.1	**512.1**	1.0	0.00	515.1	0.4	0.59	**512.1**	0.8	0.00	**512.1**	0.4	0.00
Gaskell67-32x5	562.2	**562.2**	1.0	0.00	571.9	0.6	1.73	571.9	0.8	1.73	584.6	0.6	3.98
Gaskell67-32x5 bis	504.3	**504.3**	3.0	0.00	**504.3**	0.5	0.00	534.7	1.0	6.02	504.8	0.5	0.09
Gaskell67-36x5	460.4	**460.4**	19.0	0.00	**460.4**	0.8	0.00	485.4	1.4	5.44	476.5	0.7	3.50
Min92-27x5	3062.0	**3062.0**	10.0	0.00	**3062.0**	0.4	0.00	**3062.0**	1.0	0.00	3065.2	0.3	0.11
Avg				**1.7**			**2.6**			**3.3**			**3.0**

Table 3: Solutions on Barreto's instances

[7] H. Min, V. Jayaraman, and R. Srivastava. Combined location-routing problems: a synthesis and future research directions. *European Journal of Operational Research*, 108:1–15, 1998.

[8] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31:1985–2002, 2004.

[9] C. Prins, C. Prodhon, and R. Wolfler-Calvo. A memetic algorithm with population management ($MA \mid PM$) for the capacitated location-routing problem. In J. Gottlieb and G. R. Raidl, editors, *Lecture Notes in Computer Science*, volume 3906, pages 183–194. Proceedings of EvoCOP2006 (Evolutionary Computation in Combinatorial Optimization: 6th European Conference, Budapest, Hungary, April 10-12, 2006), Springer, 2006.

[10] C. Prins, C. Prodhon, and R. Wolfler-Calvo. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR - A Quarterly Journal of Operations Research*, 4:221–238, 2006.

[11] C. Prins, C. Prodhon, and R. Wolfler-Calvo. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science*, 41(4):470–483, 2007.

[12] C. Prodhon. *Le Problème de Localisation-Routage*. PhD thesis, University of Technology of Troyes, 2006.

[13] S. Salhi and G. K. Rand. The effect of ignoring routes when locating depots. *European Journal of Operational Research*, 39:150–156, 1989.

[14] R. Srivastava. Alternate solution procedures for the locating-routing problem. *OMEGA International Journal of Management Science*, 21(4):497–506, 1993.

[15] D. Tuzun and L.I. Burke. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, 116:87–99, 1999.

[16] T.H. Wu, C. Low, and J.W Bai. Heuristic solutions to multi-depot location-routing problems. *Computers and Operations Research*, 29:1393–1415, 2002.