Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 58 (2017) 231–238 www.elsevier.com/locate/endm

A variable neighborhood search algorithm for the capacitated vehicle routing problem

Marwa Amous[a] [1] Said Toumi[a,b] [2] Bassem Jarboui[a,c] [3]
Mansour Eddaly[a,d] [4]

[a] *MODILS Lab, Université de Sfax, FSEGS, route de l'aéroport km 4, Sfax 3018 Tunisie*

[b] *Al majmaah University. College of Business Administration. Academic City, P.O Box 66, Majmaah 11952 - Kingdom of Saudi Arabia*

[c] *Emirates College of Technology, Abu Dhabi, United Arab Emirates*

[d] *Université de Gafsa, Institut Supérieur d'Administration des Entreprises, Tunisie*

Abstract

This paper studies the capacitated vehicle routing problem (CVRP). Since the problem is NP-hard, a variable neighborhood search (VNS) algorithm is proposed for the CVRP with the objective to minimize the total traveled distance. The proposed algorithm includes a variable neighborhood descent (VND) algorithm based on several different neighborhood structures to intensify the search effort. Various benchmark problems including the number of customers, the capacity of vehicles are tested to evaluate the performance of proposed methodology. The experimental results indicate that the proposed algorithm provides superior solutions for well-known benchmark problems compared to those reported in the literature.

Keywords: Vehicle routing problem, capacity constraints, variable neighborhood search, variable neighborhood descent

http://dx.doi.org/10.1016/j.endm.2017.03.030
1571-0653/

1 Introduction

The transportation problems play a key role in the supply chain management because a product is rarely produced and consumed in the same place and also reducing the transportation cost leads to enhance the performance of companies. The classical Vehicle Routing Problem (VRP) consists in determining a set of routes for an identical vehicles to serve a set of customers while minimizing the total cost of transportation. In the capacitated variant, denoted by CVRP, only capacity restrictions for vehicles are considered in addition to the basic features of the problem such that the demand of each customer, the distance between each pair of customer and the departure node (the depot). The common objective is to minimize the total cost (or length) of routes.

Over the years, several solution methods for the CVRP have been developed by researchers. Toth and Vigo [9] presented a review of models and exact approaches based on the branch and bound algorithms used to solve CVRP. The authors studied both the symmetric and asymmetric cases. The proposed exact algorithms solved the asymmetric problems with up to 300 nodes and 4 vehicles. In Mazzeo and Loiseau [7], an Ant Colony Optimization (ACO) algorithm was developed. The obtained results showed that the ACO is in competition with other metaheuristic to solve CVRP and it can find the optimal solution for the problem with less than 50 nodes. Lin et al. [6] applied hybrid algorithm of simulated annealing and tabu search. Juan et al. [5] proposed a methodology named "SR-GCWS" that combines the Clarke and WrightŠs Savings (CWS) heuristic and the Monte Carlo Simulation (MCS).

The Variable Neighborhood Search (VNS) algorithm is a metaheuristic proposed by Mladenović and Hansen [8], for solving combinatorial optimization problems, whose basic idea is the systematic change in neighborhood of a local search. The VNS algorithm has two main features; a shaking phase to escape from local optima (Diversification) and a local search phase to seek an improvement in the neighborhood of the current solution based on predefined neighborhood structures (Intensification). It uses the following findings [4]:

- a local minimum with respect to a neighborhood is not necessarily an optimal with respect to another;
- a global minimum is a local minimum with respect to all neighborhoods possible.

[1] Email: mraywaammous@hotmail.fr
[2] Email: saidtoumy@gmail.com
[3] Email: bassem_jarboui@yahoo.fr
[4] Email: eddaly.mansour@gmail.com

The Variable Neighborhood Descent (VND) is a variant of the VNS where the neighborhood structures are performed in a deterministic way [4,8].

In this paper, we propose a VNS algorithm that uses a VND in the step of the local search to expand the intensification of the search and improve as much as possible the quality of solutions.

2 Problem Formulation

Fisher and Jaikumar [2] proposed the following mathematical formulation for the Capacitated Vehicle Routing Problem (CVRP). At first, we define the required notations:

The parameters

- n: the number of customers to be visited. The customers are numbered from 1 to n and the depot has the number of 0;
- m: the number of available vehicles;
- Q: the capacity of each vehicle;
- q_i: the demand of the i^{th} customer;
- d_{ij}: the euclidean distance between customer i and j.

The decision variables

- y_i^k: a binary variable set to 1 if the command of the customer i is serviced by the vehicle k, 0 otherwise;
- x_{ij}^k: a binary variable is set to 1 if the vehicle k visits the customer i immediately before the customer j, 0 otherwise.

The integer programming formulation

$$\min \sum_{k=1}^{m} \sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} x_{ij}^k \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} y_i^k \leq Q \quad \forall k = 1, 2, ..., m \tag{2}$$

$$\sum_{k=1}^{m} y_i^k = 1 \quad \forall i = 1, 2, ..., n \tag{3}$$

$$\sum_{k=1}^{m} y_i^k = m \quad for \quad i = 0 \tag{4}$$

$$\sum_{i=0}^{n} x_{ij}^{k} = y_{i}^{k} \quad \forall j = 0, 1, ..., n; \quad \forall k = 1, 2, ..., m \tag{5}$$

$$\sum_{j=0}^{n} x_{ij}^{k} = y_{i}^{k} \quad \forall i = 0, 1, ..., n; \quad \forall k = 1, 2, ..., m \tag{6}$$

$$\sum_{i,j \in S} x_{ij}^{k} \leq |S| - 1 \quad S \subset \{1, 2, ..., n\}\,; \quad |S| \geq 2; \quad \forall k = 1, 2, ..., m \tag{7}$$

$$x_{ij}^{k} \in \{0, 1\} \quad \forall i = 0, 1, ..., n; \quad \forall j = 0, 1, ..., n; \quad \forall k = 1, 2, ..., m \tag{8}$$

$$y_{i}^{k} \in \{0, 1\} \quad \forall i = 0, 1, ..., n; \quad \forall k = 1, 2, ..., m \tag{9}$$

The objective function (1) minimizes the total distance traveled. Constraints (2) state that the vehicle capacity will not be exceeded. Constraints (3) ensure that each customer is serviced by some vehicle. Constraint (4) ensure that each route begins and ends at the depot. Constraints (5-7) are those of a Traveling Salesman Problem (TSP). Constraints (8) and (9) characterize the binary variables.

3 The proposed algorithm

The proposed VNS algorithm includes three steps: at the shaking step the algorithm tries to escape from local optimum. Then, a deterministic step is performed using a VND algorithm. Finally, an evaluation step to accept only the improved solutions according to the current one. In this section we discuss in the details the framework of the proposed algorithm.

3.1 Solution representation

A candidate solution s of the CVRP is represented by a set vectors of real numbers π_k, $k = 1, 2, ..., m$. Each vector consists of an ordered subset of customers visited by the vehicle k. Therefore, with respect to the constraints of the problem, the solution is represented as m different tours starting and ending at the depot for each vehicle, i.e., the first and the last case of each vector are both equal to zero. We note that a tour is a sequence of customers assigned to one vehicle.

3.2 Initial solution

In order to create an initial solution to the problem, we choose to use a random procedure to increase the diversification capability of the proposed algorithm.

Indeed, we select a vehicle at random. Starting from the depot, the route of the selected vehicle is progressively extended by adding randomly a non assigned customer with respect to the capacity constraint. When no customer can be added, the vehicle returns to the depot and a new route of the next vehicle will be constructed in the same way. If all vehicles are used and there remain unassigned customers, they will be randomly assigned to a vehicle where the capacity constraint is not violated. This process will be repeated until assigning all the customers.

3.3 Feasibility handling

In order to handle feasibility, the creation of infeasible solutions is usually allowed, but they are additionally penalized with a modification of the objective function. Therefore, an adaptive penalty function is used for route overcapacity and the objective function value is the total distance plus the sum of the penalties for violations of the capacity.

The penalty cost of a route π_k is given as follows: $pen\,(\pi_k) = max\,\{0, q\,(\pi_k)\}$ where $q\,(\pi_k)$ is the sum of demands of the customers serviced by the vehicle k. Indeed, the value of the objective function of a solution s can be computed as: $f\,(s) = c\,(s) + \alpha pen\,(s)$ where $c\,(s)$ is the total distance traveled by all the vehicles, $pen\,(s)$ is the sum of violations of the capacity constraints for all routes and α is a penalty parameter. This penalty function is used by [3]. The parameter α is adjusted in an adaptive way depending on the number of feasible or infeasible solutions. Initially, α is set to 1, this parameter is periodically decreased (increased) if the last h solutions were all feasible (all infeasible) with respect to the capacity constraints.

3.4 Variable Neighborhood Descent (VND) algorithm

The Variable Neighborhood Descent (VND) is the simplest variant of the basic VNS where the systematic changes of the neighborhood structures are performed in a deterministic way [4,8]. It consists in exploring several different neighborhoods until local optimum for all considered neighborhoods is reached. In our proposed VND four neighborhood structures are used:

(i) Neighborhood 1 (N_1): inserting a sequence of customers, with respect to their order, within the same tour or between two different tours.

(ii) Neighborhood 2 (N_2): inserting a sequence of customers, with reverse order, within the same tour or between two different tours.

(iii) Neighborhood 3 (N_3): swapping two customers within the same tour or between two different tours.

(iv) Neighborhood 4 (N_4): swapping two sequence of customers within the same tour or between two different tours.

The process continues within each neighborhood as long as improvements are found and stops when local optima is reached.

3.5 Shaking

The VNS includes a shaking phase to escape from a local optimum and to diversify the search by making a number of random moves. In our case, we select at random one neighborhood structure among N_1, N_2, N_3 and N_4 and k, $k \in \left\{1, 2, ..., k_{\{max\}}\right\}$, consecutive moves are performed.

4 Computational results

The proposed General Variable Neighborhood Search algorithm is evaluated on the 50 instances used by Juan et al. [5]. All instance problems are available in www.coin-or.org/SYMPHONY/branchandcut/VRP/data. The proposed method has been coded in C++ language and run on a PC with Intel Core 2 Duo CPU 2.0 GHz processor and 2.00 GB RAM memory.

In our algorithm, the penalty parameter α is adjusted every $h = 10$ iterations: if the last 10 solutions were all feasible, then α is divided by 2; if they were all infeasible, then α is multiplied by 2; otherwise, α remains unchanged.

The numerical results are displayed in Table 1. The columns "N^o", "Instance", "# nodes", "capacity", "# routes", "BKS", "CPU(BKS)", "CWS", "VNS" and "CPU (VNS)" indicate the number of the instance, the name of the instance, the number of nodes, the capacity of the vehicles, the number of routes, the best known solution, the solution obtained by the Clarke and WrightŠs Savings (CWS) heuristic of [1], the CPU time in seconds to obtain BKS, the solution obtained by our algorithm and the CPU time in seconds of our VNS algorithm. We note that BKS is the best known solution provided so far by CWS algorithm for the specified problem or by our proposed algorithm.

We observe that our proposed algorithm or is able to find the BKS or it improves them. In fact, among the 50 instance problems, 15 best known solutions are improved. Moreover, the VNS algorithms shows a significant superiority in terms of solution's quality according to the CWS heuristic.

Table 1
Computational results

N^o	Instance	# nodes	capacity	# routes	BKS	CPU(BKS)	CWS	VNS	CPU (VNS)
				set A					
1	A-n32-k5	32	100	5	787.08	6	843.69	787.08	2.142
2	A-n33-k5	33	100	5	662.11	2	712.05	662.11	1.084
3	A-n33-k6	33	100	6	742.69	3	776.26	742.69	1.097
4	A-n37-k5	37	100	5	672.47	8	707.81	672.47	2.144
5	A-n38-k5	38	100	5	733.95	7	768.14	733.95	1.334
6	A-n39-k6	39	100	6	833.2	1	863.08	833.20	2.432
7	A-n45-k6	45	100	6	944.88	31	1006.45	944.88	4.524
8	A-n45-k7	45	100	7	1146.77	45	1199.98	1146.77	4.072
9	A-n55-k9	55	100	9	1074.46	2158	1099.84	1074.46	16.162
10	A-n60-k9	60	100	9	1355.80	38	1421.88	1355.80	13.230
11	A-n61-k9	61	100	9	1039.08	47	1102.23	1039.08	30.972
12	A-n63-k9	63	100	9	1622.14	145	1687.96	1622.14	5.703
13	A-n65-k9	65	100	9	1181.69	30	1239.42	1181.69	2.448
14	A-n80-k10	80	100	10	1766.50	6877	1860.94	1766.50	54.701
				set B					
15	B-n31-k5	31	100	5	676.76	1	681.16	676.09	5.513
16	B-n35-k5	35	100	5	956.29	218	978.33	956.29	1.488
17	B-n39-k5	39	100	5	553.27	17	566.71	553.16	1.095
18	B-n41-k6	41	100	6	834.96	2	898.09	833.66	8.397
19	B-n45-k5	45	100	5	755.43	20	757.16	753.96	14.690
20	B-n50-k7	50	100	7	744.78	2	748.80	744.23	0.280
21	B-n52-k7	52	100	7	750.08	40	764.90	749.97	11.934
22	B-n56-k7	56	100	7	712.92	1348	733.74	712.92	8.095
23	B-n57-k9	57	100	9	1603.63	3	1653.42	1602.29	764.284
24	B-n64-k9	64	100	9	869.32	84	921.56	868.19	194.708
25	B-n67-k10	67	100	10	1039.36	170	1099.95	1039.27	41.974
26	B-n68-k9	68	100	9	1278.21	746	1317.77	1276.2	842.350
27	B-n78-k10	78	100	10	1228.14	2.041	1264.56	1227.9	475.454
				set E and M					
28	E-n22-k4	22	6000	4	375.28	0	388.77	375.28	0.016
29	E-n30-k3	30	4500	3	535.80	4	534.45	535.8	0.203
30	E-n33-k4	33	8000	4	838.72	7	843.10	837.67	0.078
31	E-n51-k5	51	160	5	524.61	32	584.64	524.61	0.562
32	E-n76-k7	76	220	7	687.60	887	737.74	687.60	36.364
33	E-n76-k10	76	140	10	835.26	21.707	900.26	835.26	163.504
34	E-n76-k14	76	100	14	1026.71	1853	?	1024.69	72.067
35	M-n101-k10	101	200	10	819.81	338	833.51	819.56	7.734
36	M-n121-k7	121	200	7	1043.88	74.488	1068.14	1042.11	26.370
				set P					
37	P-n19-k2	19	160	2	212.66	16	237.90	212.66	1.060
38	P-n20-k2	20	160	2	217.42	41	234.00	217.42	0.094
39	P-n22-k2	22	160	2	217.85	9	239.50	217.85	0.037
40	P-n22-k8	22	3000	8	601.42	0	590.62	600.83	0.047
41	P-n40-k5	40	140	5	461.73	3	518.37	461.73	0.764
42	P-n50-k8	50	120	8	634.85	14	674.34	634.85	31.506
43	P-n50-k10	50	100	10	699.56	379	734.32	699.56	3.338
44	P-n51-k10	51	80	10	742.48	19	790.97	741.5	4.352
45	P-n60-k10	60	120	10	748.94	166	800.20	748.07	12.948
46	P-n65-k10	65	130	10	796.67	67	851.67	795.66	8.392
47	P-n70-k10	70	135	10	829.93	52	896.86	829.93	438.808
48	P-n76-k4	76	350	4	598.19	704	689.13	598.2	42.745
49	P-n76-k5	76	280	5	635.04	73	698.51	633.32	3.306
50	P-n101-k4	101	400	4	692.64	380	765.38	692.60	600.991

5 Conclusion

In this paper a variable neighborhood search algorithm for solving the capacitated vehicle routing problem is proposed. This algorithm introduced the VND algorithm at the phase of the local search to increase the in-depth search of the VNS algorithm. The obtained results show the effectiveness of the proposed approach in improving the quality of solutions previously found in the literature.

Finally several extensions of this work can be considered such as addressing to a multi-depot network or a network that has several products. Moreover, we can consider other additional constraints (time window, delivery and pickup demand etc.).

References

[1] Clarke, G. and J. W. Wright, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations Research **12** (1964), pp. 568–581.

[2] Fisher, M. L. and R. Jaikumar, *A generalized assignment heuristic for vehicle routing*, Networks **11** (1981), pp. 109–124.

[3] Gendreau, M., A. Hertz and G. Laporte, *A tabu search heuristic for the vehicle routing problem*, Management Science **40** (1994), pp. 1276–1290.

[4] Hansen, P. and N. Mladenović, *Variable neighborhood search: Principles and applications*, European Journal of Operational Research **130** (2001), pp. 449–467.

[5] Juan, A. A., J. Faulin, R. Ruiz, B. Barrios and S. Caballe, *The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem*, Applied Soft Computing **10** (2010), pp. 215–224.

[6] Lin, S.-W., K.-C. Lee, Z-J.and Ying and C.-Y. Lee, *Applying hybrid metaheuristics for capacitated vehicle routing problem*, Expert Systems with Applications **36** (2009), pp. 1505–1512.

[7] Mazzeo, S. and I. Loiseau, *An ant colony algorithm for the capacitated vehicle routing*, Electronic Notes in Discrete Mathematics **18** (2004), pp. 181–186.

[8] Mladenović, N. and P. Hansen, *Variable neighborhood search*, Computers & Operations Research **24** (1997), pp. 1097–1100.

[9] Toth, P. and D. Vigo, *Models, relaxations and exact approaches for the capacitated vehicle routing problem*, Discrete Applied Mathematics **123** (2002), pp. 487–512.