

A novel heuristic algorithm for capacitated vehicle routing problem

Sena Kır¹ · Harun Reşit Yazgan¹ · Emre Tüncel¹

Received: 10 May 2016 / Accepted: 2 February 2017 / Published online: 15 February 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract The vehicle routing problem with the capacity constraints was considered in this paper. It is quite difficult to achieve an optimal solution with traditional optimization methods by reason of the high computational complexity for large-scale problems. Consequently, new heuristic or metaheuristic approaches have been developed to solve this problem. In this paper, we constructed a new heuristic algorithm based on the tabu search and adaptive large neighborhood search (ALNS) with several specifically designed operators and features to solve the capacitated vehicle routing problem (CVRP). The effectiveness of the proposed algorithm was illustrated on the benchmark problems. The algorithm provides a better performance on large-scaled instances and gained advantage in terms of CPU time. In addition, we solved a real-life CVRP using the proposed algorithm and found the encouraging results by comparison with the current situation that the company is in.

Keywords Capacitated vehicle routing problem (CVRP) · Tabu search · Adaptive large neighborhood search (ALNS)

Introduction

The vehicle routing problem with capacity constraints is one of the most important subjects for logistic activities. The main objective of vehicle routing problem is to design

the least cost routes for a fleet of vehicles from one depot to a set of points. Vehicles belong to a fleet and located to the central depot and have a capacity. Each customer, which located a point to visit, has amount of demand. The cost depends on the distance. Each route starts from the central depot and finish at the central depot and the vehicle capacity must not be overloaded.

There are several studies about CVRP in the literature over the past decade and heuristic, metaheuristic or hybrid methods were studied overwhelmingly. Mazzeo and Loiseau (2004) developed an ant colony algorithm. They resumed the main characteristics and tried several alternatives for each component of the algorithm. The proposed algorithm was compared with other heuristics on the literature in a different set of problems and the effectiveness of the algorithm was illustrated. Fukasawa et al. (2006) proposed an algorithm based on branch and bound cut and traditional Lagrangian relaxation over q routes. The algorithm was tested on more than a hundred of instances. Mester and Brasys (2007) presented an adaptation of active-guided evolution strategies metaheuristic. The proposed metaheuristic approach is the combination of guided local search and evolution strategies metaheuristics. It was carried out on a set of instances and the results illustrated the proposed approach is highly competitive. Lin et al. (2009) developed a hybrid algorithm which takes the advantages of simulated annealing and tabu search. The developed algorithm included classical large-scaled CVRP instances to verify the effectiveness. Ai and Kachitvichyanukul (2009) presented $(n + 2m)$ - and $(3m)$ -dimensional particle swarm optimization with n customers and m vehicles. $(n + 2m)$ starts with the transformation of particle into a priority list of customer to enter route and priority matrix, $3m$ starts with the transformation of particle into the vehicle orientation points and the vehicle

✉ Harun Reşit Yazgan
yazgan@sakarya.edu.tr

Sena Kır
senas@sakarya.edu.tr

¹ Department of Industrial Engineering, Sakarya University, Sakarya, Turkey

coverage radius. The computational results illustrated (3 *m*)-dimensional particle swarm is better than other. Yurtkuran and Emel (2010) proposed an electromagnetism-like algorithm which is hybridized with a local search method. They stated that the algorithm was straightforward, easy to use and efficient for CVRP. Szeto et al. (2011) proposed an artificial bee colony algorithm and compared with the other methods in the literature. They asserted that the proposed algorithm is an alternative to solve the CVRP. Nazif and Lee (2012) presented a genetic algorithm using optimized crossover operator and showed the effectiveness of the algorithm on the benchmark instances. Cui et al. (2013) developed a new improved quantum evolution algorithm with a mixed local search procedure and illustrated the effectiveness of the algorithm in different cases. Kao and Chen (2013) developed a two-stage hybrid algorithm that integrated a discrete particle swarm optimization and simulated annealing. The algorithm can decrease the computational time in consequence of eliminating the number of infeasible solutions. Jin et al. (2014) presented a cooperative parallel metaheuristic which consists of multiple parallel tabu search threads that cooperate by asynchronously exchanging best found solutions through a common solution pool. They compared the algorithm with the others in the literature and the computational results illustrated the effectiveness and competitiveness of the algorithm. LinHui and Luo (2014) developed a novel fusion algorithm which is a dynamic adaptive immune genetic algorithm with automatic immune monitoring function. They proved success of the proposed algorithm on the small-scale test problems. Mazidi et al. (2016) focused on enhancing the capability of local search algorithms. In this context, they compared six different metaheuristic algorithms (simulated annealing, stochastic hill climbing, ant colony, tabu search, particle swarm optimization, genetic algorithm) and proposed an improved genetic algorithm which uses ant colony algorithm as the initial population creating operator. Yeh et al. (2016) proposed a two-stage iterated local search strategy based on random variable neighbor descent and showed the superiority from the iterated local search. Akpınar (2016) developed a large neighborhood search algorithm which is hybridized with the ant colony optimization algorithm as the solution construction mechanism. They showed the algorithm has satisfactory results on instances.

As exemplified above, there are several studies about heuristics and metaheuristic approach to solve large-scaled CVRP because of NP-hard structure of the problem. In this paper, we developed a new heuristic algorithm based on tabu search and ALNS. Our proposed algorithm is inspired Ropke and Pisinger (2006) in terms of ALNS but we brought three important theoretical contributions and achieve a good efficiency for our problem. The first one is

the design of specific destroy/repair operators, which considers both the sequence of the routes and vehicle capacity. The second modification is about the search operator which works to cluster service points. The third one is the design of a diversification technique which depends on the clusters. The proposed algorithm was tested on well-known different instances. The computational experiments illustrate that our heuristic algorithm is competitive with other proposed algorithms for solving CVRP.

The remainder of this paper is organized as follows. In the next section, a brief overview of the vehicle routing problem is given followed by which a new proposed algorithm is explained. The subsequent section deals with computational results on benchmark problems. Results of a numerical investigation which show the comparison of the proposed algorithms on a real-life example are presented before the concluding section. Finally, conclusions are presented.

Vehicle routing problem

The vehicle routing problem (VRP) consists of the problems in which a set of routes for a fleet of vehicles based at one or several depots must be determined for a number of geographically dispersed cities or customers. The aim of the classical VRP is finding a set of routes at a minimal cost (finding the shortest path, minimizing the number of vehicles, etc.) beginning and ending the route at the depot, so that the known demand of all nodes are fulfilled. In the literature, VRP is commonly defined under capacity constraints, so the VRP generally has the same meaning with CVRP. The mathematical formulation of the CVRP is shown below:

A capacity of each vehicle

V maximum number vehicle

F_{ij} the flow of cores from node *i* to *j*

Z total transportation cost

d_i demand at node *i*

c_{ij} Distance between node *i* and node *j*

$$x_{ij} = \begin{cases} 1, & \text{if vehicle moves from node } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

N number of nodes

The objective function can be written as follows:

$$\text{Min } Z = \sum_{i=1}^N \sum_{j=1, j \neq i}^N c_{ij} \cdot x_{ij} \quad (1)$$

The model constraints are:

$$\sum_{l=2, k \neq l}^N x_{lk} + x_{1k} = 1 \quad \forall k \quad (2)$$

$$\sum_{l=2, k \neq l}^N x_{kl} + x_{k1} = 1 \quad \forall k \tag{3}$$

$$\sum_{k=2}^N x_{1k} \leq V \tag{4}$$

$$\sum_{j=1, j \neq i}^N x_{ij} = \sum_{j=1, j \neq i}^N x_{ji} \quad \forall i \tag{5}$$

$$x_{kk} = 0 \quad \forall k \tag{6}$$

$$x_{lk} + x_{kl} = 1 \quad \forall k, l, k \neq l \tag{7}$$

$$\sum_{j=1, j \neq i}^N F_{ij} = \sum_{j=1, j \neq i}^N F_{ji} + d_i \quad \forall i_{i > 1} \tag{8}$$

$$d_i \cdot x_{ij} \leq F_{ij} \quad \forall i, j, i \neq j \tag{9}$$

$$F_{ij} \leq (A - d_j) \cdot x_{ij} \quad \forall i, j, i \neq j \tag{10}$$

The objective function (1) of the model is to minimize the total traveled distance. The constraints (2) and (3) state that there is exactly one departure from node i . The constraint (4) provides not to exceed the total number of vehicles. Constraint (5) provides balance between incoming and outgoing arcs at a given node. Constraint (6) eliminates flow from node i to node i . Constraint (7) is trivial sub-tour elimination constraint. Constraints (8), (9) and (10) provide balance between total inflow and outflow at node.

This mathematical model can be used for only small-scale CVRP because it is quite difficult to achieve an optimal solution with traditional optimization methods due to the high computational complexity for large-scale problems. For this reason, we constructed a novel heuristic algorithm for CVRP in this study.

A novel algorithm for CVRP

In this section, our proposed algorithm is explained. We used following notation in the algorithm. The points to visit are denoted by a set of S which has number of p elements as $S = \{1, \dots, p\}$. All the points are also denoted analytically by an arc set of P as $P = \{(x_i, y_i) : i \in S\}$. $\Delta_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)^{0.5}$ is the distance matrix which states the distance between the elements of S set. A solution is the sets R of k routes and also vehicles, $R_k \subseteq S$ and k are less than and equal to the number of free vehicle t . All the points to visit have a demand d_i and the capacity of free vehicles meeting the total demand is assumed as $\sum d_i \leq tC$. We

determined the objective function as $F = \sum_{k=1}^t \sum_{i=1}^p \sum_{j=1}^p \pi_{ijk} \Delta_{ij}$ for all $i, j \in S, i \neq j$ and $k \leq t$, where

$$\pi_{ijk} = \begin{cases} 1 & \text{from } i \text{ to } j \text{ by vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

The capacity constraints must be kept down while searching for a solution. Therefore, we determined a function to control the feasibility of the candidate solution as $\sum_{k=1}^t \pi_{ijk} d_j \leq C$ for all $i, j \in S$ and $i \neq j$.

One of our motivations is that the quality of initial solution affected the performance of the algorithm. Hence, we prefer to start the solution by these steps:

Step 1: Start from a random i point and determine the π_{ijk} as 1 considering the minimum value of Δ_{ij} , where the $S = S^*$ and $R_k = R_k^*$.

Step 2: Calculate the $\bar{C}_k = C_k + d_j$ and $S^* = S \setminus \{j\}$ and $R_{k^*} = R_k \cup \{j\}$.

Step 3: Repeat steps 1 and 2 until $\bar{C}_k \geq C$. If this is met, increase the value of k as 1.

Step 4: Run the all previous steps until $S^* \in \emptyset$. At last all of R_{k^*} state us the solution.

After determining the initial solution, new solutions are determined by ALNS. In this study, we constructed a relocation procedure to search neighborhoods. The steps of the procedure consist of these steps:

Stage 1: Select one of each point i from each routes R_{k^*} by roulette wheel principle using Eq. (11).

$$p_k = \frac{\sum_{i=1}^p \sum_{j=1}^p \pi_{ijk} \Delta_{ij}}{\sum_{k=1}^t \sum_{i=1}^p \sum_{j=1}^p \pi_{ijk} \Delta_{ij}} \tag{11}$$

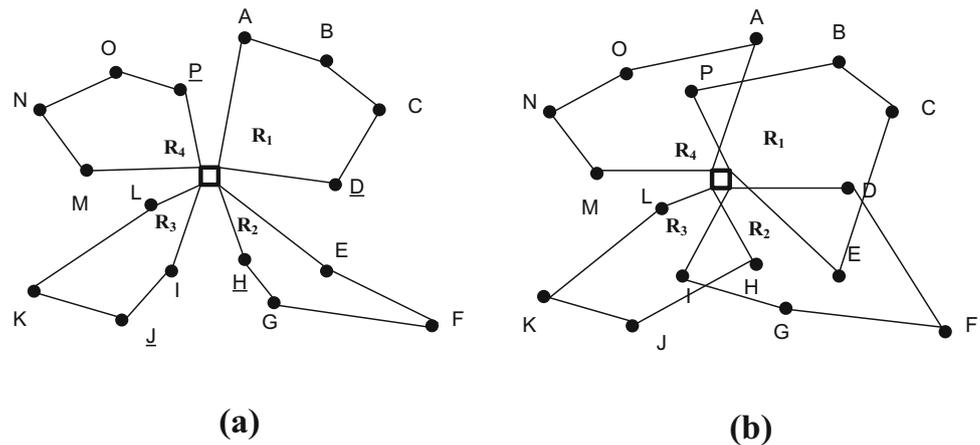
Stage 2: Determine the second closest points to each selected points according to the Δ_{ij} matrix.

Stage 3: Start from the R_1 and relocate the selected point with its second closest point. If the selected point and its second point are in the same route, do not relocate them. So we can prevent to enhance the solution space too much.

Stage 3': While relocating, $\bar{C}_k = C_k + d_j$ should be revised and route capacity should be controlled.

Stage 4: Stages 2 and 3 are repeated for all the points which were selected in Stage 1.

The relocation procedure is illustrated in Fig. 1. In the part (a) of Fig. 1D, H, J and P points were randomly selected. The second closest points to them E, I, H and A

Fig. 1 Relocation**Fig. 2** Pseudo codes of proposed algorithm

```

while ( $F = F_0$  and  $k = 1$ )
   $F^* = F_k$ 
  if (not contains ( $F^*$ , tabu list) and not contains (movement list)) or ( $F^* > F_{best}$ )
     $c(F^*) = k \times$  capacity violation
    if ( $c(F^*) < c(F)$ )
      set  $F = F^*$  and  $c(F) = c(F^*)$ 
      Update (frequency matrix, tabu list, movement list, movement matrix,
      k)
    end
  if ( $F > F_{best}$  and  $s$  is feasible)
     $F_{best} = F$ 
  end
end
return ( $F_{best}$ )

```

were chosen respectively. Part (b) of Fig. 1 shows us a neighbor solution after relocation under condition of the capacity constraint is not exceeded.

In addition, there is a movement list to prevent cycling in the proposed algorithm. This list prohibits the use of some specific set of solutions for several iterations if one of them is used in recent iterations. Therefore, the search does not travel around the local optimum points. We preferred to determine the size of the list experimentally.

Finally, each iteration of the proposed algorithm that can be achieved by serving two vertices on the same route instead of leaving them on separate routes is implemented to improve the solution and a non-tabu solution is selected as a new solution. The algorithm after determining an initial solution steps can be summarized in Fig. 2 by pseudocodes.

Computational results on benchmark problems

The proposed algorithm is tested on three (A, M and G) sets of benchmark instances with different number of customers/points in the literature. Data of the problems and

results were taken from <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/>. Table 1 illustrates the comparison between the optimal/best objective values of our best objective values in the literature for these benchmark instances.

The proposed heuristic algorithm can find the optimal solution immediately for small-scale problems. We mean with small scale that which has less than fifty points. When the scale of the problem gets larger, the algorithm converged to the optimal solution with 0.01% difference approximately. In addition to this, we found the new best solutions for the problems whose optimal solutions have not been found yet. The new best routes are shown in “Appendix 1” for 28th, 29th and 30th problems of Table 1.

A real-life application

In this section, we will summarize a project which motivated us to study this problem. We studied on the CVRP problem of a company which produces office furniture in Sakarya City, Turkey. The company produces and distributes the office furniture products (staff tables, meeting



Table 1 Comparison between the solutions of proposed algorithm and best well-known solutions

Prob. #	Prob. code	Optimal/ Best	Current result	Our result	Difference	% Difference	CPU time (s)
1	A-n32-k5	Optimal	784	784	0	0.000	1.26
2	A-n33-k5	Optimal	661	661	0	0.000	1.32
3	A-n33-k6	Optimal	742	742	0	0.000	2.28
4	A-n34-k5	Optimal	778	778	0	0.000	2.98
5	A-n36-k5	Optimal	799	799	0	0.000	7.58
6	A-n37-k5	Optimal	669	669	0	0.000	12.65
7	A-n37-k6	Optimal	949	949	0	0.000	14.89
8	A-n38-k5	Optimal	730	730	0	0.000	22.32
9	A-n39-k5	Optimal	822	822	0	0.000	48.50
10	A-n39-k6	Optimal	831	831	0	0.000	44.62
11	A-n44-k6	Optimal	937	939	2	0.002	102.03
12	A-n45-k6	Optimal	944	955	11	0.012	97.88
13	A-n45-k7	Optimal	1146	1153	7	0.006	230.56
14	A-n46-k7	Optimal	914	915	1	0.001	201.93
15	A-n48-k7	Optimal	1073	1073	0	0.000	298.76
16	A-n53-k7	Optimal	1016	1026	10	0.010	1987.48
17	A-n54-k7	Optimal	1167	1169	2	0.002	2369.52
18	A-n55-k9	Optimal	1073	1074	1	0.001	3892.10
19	A-n60-k9	Optimal	1354	1366	12	0.009	7760.19
20	A-n61-k9	Optimal	1034	1045	11	0.011	7239.88
21	A-n62-k8	Optimal	1288	1302	14	0.011	12331.62
22	A-n63-k9	Optimal	1616	1644	28	0.017	11906.02
23	A-n63-k10	Optimal	1314	1325	11	0.008	11999.34
24	A-n64-k9	Optimal	1401	1442	41	0.029	17002.33
25	A-n65-k9	Optimal	1174	1189	15	0.013	18071.70
26	A-n69-k9	Optimal	1159	1169	10	0.009	18720.03
27	A-n80-k10	Optimal	1763	1790	27	0.015	20692.47
28	M-n151-k12	Best	1053	1048	-5	-0.005	37360.19
29	M-n200-k17	Best	1373	1331	-42	-0.031	42873.25
30	G-n262-k25	Best	6119	5875	-244	-0.04	45202.09

Table 2 Comparison between our solutions and company’s solutions for 30 days

No.	# of customer	# of vehicle	Company’s solutions (km)	Our solutions (km)	No.	# of customer	# of vehicle	Company’s solutions (km)	Our solutions (km)
1	105	8	1732	1647	16	115	8	1819	1722
2	117	8	1912	1880	17	109	8	1752	1740
3	124	10	2601	2509	18	92	7	1057	1043
4	123	11	2405	2401	19	98	7	1204	1107
5	104	8	1427	1292	20	97	7	1187	1092
6	95	7	1176	1095	21	40	3	648	637
7	101	8	1397	1303	22	106	8	1460	1359
8	79	6	994	967	23	116	8	1843	1739
9	118	9	1926	1926	24	109	8	1633	1581
10	101	8	1320	1267	25	102	8	1335	1275
11	127	11	2731	2682	26	128	10	2691	2543
12	121	9	2541	2390	27	130	10	2978	2810

Table 2 continued

No.	# of customer	# of vehicle	Company's solutions (km)	Our solutions (km)	No.	# of customer	# of vehicle	Company's solutions (km)	Our solutions (km)
13	120	9	2513	2441	28	132	9	2990	2903
14	99	7	1050	1012	29	114	8	1868	1769
15	110	8	1861	1798	30	106	8	1417	1301

Table 3 Paired samples statistics

	<i>N</i>	Mean	St dev	SE mean
Company's solutions	30	1782	638	117
Our solutions	30	1708	623	114
Difference	30	74.57	45.28	8.27

95% CI for mean difference: (57.66; 91.47)

T test of mean difference = 0 (vs not = 0): *T* value = 9.02
P value = 0.000

tables, workstations, working chairs, guest chairs, etc.) to the customers. The company has sufficient number of identical vehicles for carrying. The vehicles distribute orders from 09:00 a.m. to 18:00 p.m. That is to say, there are only 9 h for distribution and assembly per day. Based on this, the vehicles can be loaded up to 15 customers' orders. In addition, the weight capacity is 1500 kg and volumetric load capacity is 5500 L of the vehicles.

We handled the 30-day transportation problem of the company and we solved these 30 different problems and run up to 21,600 s by the proposed algorithm and compared with company's previous solutions. Then we applied paired sample *t* test to evaluate the performance of these solutions to illustrate the efficiency of our algorithm.

T test is applied to compare the solutions illustrated in Table 2. It illustrates that the company's solutions and the proposed algorithm's solutions are different in a given confidence interval and proposed algorithm's solutions are better than company's solutions.

$$H_0 = \mu_A = \mu_B$$

$$H_1 = \mu_A \neq \mu_B$$

where μ_A average cost of 30 problems solved by the company μ_B average cost of 30 problems solved by the proposed algorithm.

First hypothesis is average cost of 30 problems solved by the company is equal to average cost of 30 problems solved by proposed algorithm. Second hypothesis is average cost of 30 problems solved by the company is not equal to average cost of 30 problems solved by proposed algorithm. The paired samples statistics are shown in Table 3.

According to the *p* value in Table 3, hypothesis H_0 is rejected. Because it is less than 0.05 in 95% confidence interval. This means that there is strong evidence that

these two solutions are different. Because of the positive of the value, company's solutions are worse than our solutions. In conclusion, this test shows that the solutions of our algorithm decrease the transportation cost of the company.

Conclusion

In this paper, we considered capacitated vehicle routing problem. There are various studies done and tested on this problem. These studies highly succeeded in studying this NP-hard problem. Some features of tabu search and ALNS providing this success are: allowing infeasible solutions, flexible parameters, destroy/repair operators, diversification strategy, intensification strategy, and adaptive memory. To this respect, we proposed a new heuristic algorithm based on tabu and ALNS for CVRP. We tested the proposed algorithm and the algorithm provides compatible results on benchmark instances. For three of benchmark instances, the proposed algorithm generated better solutions compared with best well-known solutions in the literature. In the first 11 benchmark instances, our algorithm found optimal solutions. However, it converged to the optimal solution with 0.01% difference approximately in the case of not to find optimal solution. Finally, when we applied the algorithm on a real-life problem the results satisfied us in terms of effectiveness and productiveness. Using the proposed algorithm substituted for the current method provides advantage to the company because the results are better than current methods and CPU time is allowable.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix 1

See Table 4.

Table 4 New solution of the three well-known instances

Problem # 28	Route # 1: 0, 7, 19, 107, 11, 64, 49, 143, 36, 47, 124, 46, 8, 114, 18, 0
	Route # 2: 0, 122, 20, 128, 66, 71, 65, 136, 35, 135, 34, 78, 129, 79, 3, 77, 0
	Route # 3: 0, 92, 98, 91, 141, 16, 86, 140, 38, 14, 119, 44, 100, 37, 0
	Route # 4: 0, 53, 40, 21, 73, 72, 74, 23, 67, 39, 139, 4, 110, 0
	Route # 5: 0, 58, 137, 144, 57, 15, 43, 142, 42, 87, 97, 95, 117, 13, 0
	Route # 6: 0, 2, 115, 145, 41, 22, 133, 75, 56, 25, 55, 130, 54, 149, 26, 105, 0
	Route # 7: 0, 28, 76, 116, 68, 121, 29, 24, 134, 80, 150, 109, 12, 138, 0
	Route # 8: 0, 147, 60, 83, 125, 45, 17, 113, 61, 85, 93, 96, 6, 0
	Route # 9: 0, 111, 50, 102, 33, 81, 120, 9, 103, 51, 1, 132, 0
	Route # 10: 0, 27, 127, 31, 10, 108, 126, 63, 90, 32, 131, 30, 70, 101, 69, 0
	Route # 11: 0, 89, 118, 84, 5, 99, 104, 59, 94, 112, 0
	Route # 12: 0, 146, 88, 148, 62, 123, 48, 82, 106, 52, 0
Problem # 29	Route # 1: 0, 154, 12, 150, 80, 68, 116, 184, 111, 0
	Route # 2: 0, 82, 48, 123, 19, 107, 175, 11, 159, 62, 148, 0
	Route # 3: 0, 146, 52, 153, 106, 194, 7, 182, 88, 190, 127, 167, 27, 0
	Route # 4: 0, 198, 110, 4, 155, 139, 187, 39, 75, 133, 22, 41, 145, 0
	Route # 5: 0, 50, 102, 157, 185, 79, 129, 169, 29, 121, 77, 196, 76, 28, 0
	Route # 6: 0, 1, 51, 161, 71, 65, 136, 35, 135, 164, 34, 78, 158, 3, 0
	Route # 7: 0, 138, 109, 177, 163, 24, 134, 54, 195, 26, 105, 0
	Route # 8: 0, 31, 126, 63, 181, 64, 49, 143, 36, 47, 168, 124, 46, 0
	Route # 9: 0, 156, 18, 114, 8, 174, 45, 125, 199, 83, 60, 166, 89, 0
	Route # 10: 0, 137, 144, 172, 42, 142, 14, 140, 38, 119, 192, 100, 98, 37, 95, 94, 0
	Route # 11: 0, 53, 180, 21, 72, 74, 171, 115, 178, 2, 152, 58, 0
	Route # 12: 0, 147, 5, 173, 61, 16, 86, 113, 17, 84, 118, 0
	Route # 13: 0, 112, 183, 96, 104, 99, 59, 151, 92, 97, 13, 0
	Route # 14: 0, 149, 179, 130, 165, 55, 25, 170, 67, 23, 186, 56, 197, 73, 40, 0
	Route # 15: 0, 132, 69, 162, 10, 189, 108, 90, 32, 131, 160, 30, 70, 101, 0
	Route # 16: 0, 176, 122, 128, 20, 188, 66, 103, 9, 120, 81, 33, 0
	Route # 17: 0, 6, 93, 85, 193, 91, 191, 141, 44, 43, 15, 57, 87, 117, 0
Problem # 30	Route # 1: 0, 144, 218, 119, 94, 151, 86, 149, 247, 88, 50, 77, 93, 200, 0
	Route # 2: 0, 14, 117, 98, 240, 107, 212, 45, 91, 46, 0
	Route # 3: 0, 158, 60, 110, 41, 210, 20, 109, 227, 29, 75, 237, 0
	Route # 4: 0, 115, 244, 99, 70, 108, 155, 76, 154, 87, 188, 256, 4, 164, 0
	Route # 5: 0, 252, 234, 68, 142, 232, 231, 202, 139, 1, 229, 71, 31, 205, 211, 5, 204, 0
	Route # 6: 0, 118, 44, 22, 120, 128, 143, 174, 196, 48, 23, 236, 111, 0
	Route # 7: 0, 140, 160, 28, 182, 95, 34, 81, 186, 172, 0
	Route # 8: 0, 165, 222, 85, 257, 253, 92, 178, 105, 97, 132, 52, 84, 0
	Route # 9: 0, 74, 7, 156, 259, 35, 239, 184, 192, 152, 208, 246, 0
	Route # 10: 0, 233, 67, 51, 134, 159, 147, 245, 40, 127, 24, 258, 122, 0
	Route # 11: 0, 243, 169, 171, 181, 59, 162, 219, 0
	Route # 12: 0, 102, 15, 136, 78, 201, 32, 226, 176, 56, 242, 0
	Route # 13: 0, 254, 66, 175, 17, 55, 47, 168, 103, 113, 194, 80, 42, 197, 0
	Route # 14: 0, 255, 228, 53, 124, 49, 130, 129, 221, 90, 61, 12, 133, 79, 261, 0
	Route # 15: 0, 148, 26, 9, 153, 189, 106, 121, 248, 89, 167, 13, 251, 0
	Route # 16: 0, 213, 21, 100, 145, 209, 138, 33, 72, 166, 96, 0
	Route # 17: 0, 135, 16, 6, 37, 25, 141, 104, 215, 0
	Route # 18: 0, 43, 223, 38, 39, 146, 3, 123, 190, 0
	Route # 19: 0, 8, 126, 250, 180, 63, 216, 161, 187, 179, 199, 116, 0
	Route # 20: 0, 2, 170, 230, 58, 150, 183, 114, 0
	Route # 21: 0, 235, 225, 241, 11, 10, 238, 131, 214, 57, 203, 0
	Route # 22: 0, 27, 260, 173, 137, 18, 30, 191, 19, 185, 207, 0
	Route # 23: 0, 195, 36, 220, 65, 206, 83, 217, 0
	Route # 24: 0, 112, 73, 198, 54, 0
	Route # 25: 0, 125, 101, 193, 82, 157, 62, 249, 177, 69, 0
	Route # 26: 0, 163, 224, 64, 0

References

- Ai TJ, Kachitvichyanukul V (2009) Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Comput Ind Eng* 56:380–387. doi:[10.1016/j.cie.2008.06.012](https://doi.org/10.1016/j.cie.2008.06.012)
- Akpinar S (2016) Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. *Expert Syst Appl* 61:28–38. doi:[10.1016/j.eswa.2016.05.0230957-4174](https://doi.org/10.1016/j.eswa.2016.05.0230957-4174)
- Cui L, Wang L, Deng J, Zhang J (2013) A new improved quantum evolution algorithm with local search procedure for capacitated vehicle routing problem. *Math Probl Eng*. doi:[10.1155/2013/159495](https://doi.org/10.1155/2013/159495)
- Fukasawa R, Longo H, Lysgaard J, Aragao MP, Reis M, Uchoa E, Werneck RF (2006) Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math Program Ser A* 106:491–511. doi:[10.1007/s10107-005-0644-x](https://doi.org/10.1007/s10107-005-0644-x)
- Jin J, Crainic TG, Lokketangen A (2014) A cooperative parallel metaheuristic for capacitated vehicle routing problem. *Comput Oper Res* 44:33–41. doi:[10.1016/j.cor.2013.10.004](https://doi.org/10.1016/j.cor.2013.10.004)
- Kao Y, Chen M (2013) Solving the CVRP Problem Using a Hybrid PSO Approach, ISBN: 978-3-642-35637-7. *Comput Intell* 465:59–67. doi:[10.1007/978-3-642-35638-4](https://doi.org/10.1007/978-3-642-35638-4) (Springer, Heidelberg)
- Lin S-W, Lee Z-J, Ying K-C, Lee C-Y (2009) Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Syst Appl* 36:1505–1512. doi:[10.1016/j.eswa.2007.11.060](https://doi.org/10.1016/j.eswa.2007.11.060)
- LinHui C, Luo Z (2014) An Improved Immune Genetic Algorithm for Capacitated Vehicle Routing Problem. *The Open Cybernetics & Systemics Journal* 8:560–565. doi:[10.2174/1874110X01408010560](https://doi.org/10.2174/1874110X01408010560)
- Mazidi A, Fakhrahmad M, Sadreddini M (2016) A meta-heuristic approach for CVRP problem: local search optimization based on GA and ant colony. *J Adv Comput Res* 7:1–22
- Mazzeo S, Loiseau I (2004) An ant colony algorithm for the capacitated vehicle routing. *Electron Notes Discret Math* 18:181–186. doi:[10.1016/j.endm.2004.06.029](https://doi.org/10.1016/j.endm.2004.06.029)
- Mester D, Braysy O (2007) Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Comput Oper Res* 34:2964–2975. doi:[10.1016/j.cor.2005.11.006](https://doi.org/10.1016/j.cor.2005.11.006)
- Nazif H, Lee LS (2012) Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Appl Math Model* 36:2011–20117. doi:[10.1016/j.apm.2011.08.010](https://doi.org/10.1016/j.apm.2011.08.010)
- Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp Sci* 40(4):455–472
- Szeto WY, Wu Y, Ho SC (2011) An artificial bee colony algorithm for the capacitated vehicle routing problem. *Eur J Oper Res* 215:126–135. doi:[10.1016/j.ejor.2011.06.006](https://doi.org/10.1016/j.ejor.2011.06.006)
- Yeh CC, Lui DY, Liao YK (2016) Two-stage iterated local search for solving capacitated vehicle routing problems. *Int Symp Comput Consum Control* ISBN: 9781509030729
- Yurtkuran A, Emel E (2010) A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. *Expert Syst Appl* 37:3421–3433. doi:[10.1016/j.eswa.2009.10.005](https://doi.org/10.1016/j.eswa.2009.10.005)

