



A Hybrid Brain Storm Optimization Algorithm for Dynamic Vehicle Routing Problem

Mingde Liu, Yang Shen, and Yuhui Shi^(✉)

Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen 518055, China
{11849245,sheny3}@mail.sustech.edu.cn, shiyh@sustech.edu.cn

Abstract. The Dynamic Vehicle Routing Problem (DVRP) has many real-world applications and practical values. The objective of DVRP is to find the optimal routes for a fleet of vehicles to service the given customer requests, without violating the vehicle capacity constraint. In this paper, a hybrid algorithm is proposed for solving the DVRP with the objective to minimize the total distance of the vehicles. The Brain Storm Optimization in objective space (BSO-OS) is applied to guide the choice of different strategies for the periodic reoptimization of routes. In the BSO-OS procedure, Adaptive Large Neighborhood Search (ALNS) and Ant Colony System (ACS) are used to generate new solutions. The experiments on the DVRP benchmark and comparative studies are conducted, from which 12 out of 21 new best solutions are obtained by the proposed algorithm, and the other nine solutions are also very competitive. The experimental results show that the proposed algorithm is very effective and competitive.

Keywords: Brain storm optimization in objective space · Ant colony system · Adaptive Large Neighborhood Search · Dynamic vehicle routing problem

1 Introduction

In recent years, logistics has drawn considerable attention in both research and industry. The Vehicle Routing Problem (VRP) is a classic logistics problem that seeks to find the optimal set of routes for a fleet of vehicles to service a given set of customers. Generally, the objective function of the VRP is to minimize the total travel distance. The VRP is proved NP-hard [10], when the size of the problem increases, exact algorithms become very time consuming and maybe invalid [5]. Therefore, researchers tend to use heuristics to solve VRPs.

In real-world scenarios, logistics companies or ride-hailing companies are facing the Dynamic Vehicle Routing Problems (DVRPs) [9]. For DVRPs, new customer requests arrive dynamically when the vehicles have already started servicing existing customer requests. Therefore, the routes of the vehicles have to

be replanned at run time to minimize the cost, i.e, the routes have to be reoptimized. The reoptimization of DVRPs can be classified into two categories: periodic reoptimization and continuous reoptimization [14]. Generally, periodic reoptimization methods start with a first optimization of the initially known customer requests, then an optimization procedure solves the static VRP periodically with a fixed time interval [8]. For continuous reoptimization, whenever the new customer requests arrive, the reoptimization procedure starts to solve the new incoming data to update the routes [20]. Continuous reoptimization requires extensive computational resources. Also, if the new customer requests arrive very frequently, continuous reoptimization methods may repeat running and fail to update the routes. The advantages of periodic reoptimization are: 1) it transforms DVRPs to static VRPs to solve, and there exists extensive research on static VRPs; 2) the reoptimization is run independently during each time interval; 3) computational resources would be enough for each time interval; 4) it is more similar to practical applications. In this paper, we address the periodic reoptimization of DVRPs.

Many researches have been carried out based on the benchmark for DVRPs proposed by Kilby et al. [8] and extended by Montemanni et al. [12]. An algorithm based on Ant Colony System (ACS) was first proposed by Montemanni et al. [12] for the DVRPs. The ACS makes use of the pheromone to track the good components of a good solution. When the next time interval comes, the good components can be used by the reoptimization procedure. In addition, Hanshar et al. applied Genetic Algorithm (GA) to solve the DVRP and compared it with Tabu Search (TS) [6]. The proposed DVRP-GA algorithm used inversion operator [11] for mutation and a problem-specific operator Best-Cost Route Crossover (BCRC) for crossover. Moreover, a comparative study was carried out between the Dynamic Adapted Particle Swarm Optimization (DAPSO) algorithm and the Variable Neighborhood Search (VNS) algorithm for the DVRPs [7]. In the comparative study, the proposed DAPSO algorithm uses adaptive memory to reuse the information from the previous solutions, and the VNS algorithm systematically changes neighborhoods to escape from local optima. Additionally, an enhanced GA-based system [1] for DVRP was proposed to improve the DVRP-GA. The main modifications include the initial population of the time slices, the selection process, swap mutation, and a Local Optimal Condition (LOC) detection and escape strategy.

Brain Storm Optimization (BSO) [16,18] was first proposed in 2011, which is inspired by the human brainstorming process. In DVRPs, choosing different strategies to optimize the current state is similar to the brainstorming process. In the BSO procedure, different strategies were applied for simulating the problem owners to choose good solutions that they believe in the brainstorming process. To reduce the computational cost, an enhanced BSO algorithm in objective space named BSO-OS [17] was proposed in 2015. In this paper, we applied the BSO-OS algorithm and proposed a hybrid BSO algorithm named BSO-DVRP for solving DVRPs. The overall frame of the algorithm is BSO, and in the BSO procedure,

two popular VRP heuristics are applied to generate new solutions, which are Adaptive Large Neighborhood Search (ALNS) [15] and ACS [3].

The rest of this paper is organized as follows. Section 2 illustrates the definition and model of DVRP. Section 3 introduces the proposed BSO-DVRP algorithm. Section 4 first describes the benchmark and then evaluates the proposed algorithm. Section 5 concludes the paper.

2 Problem Definition

In this paper, we explore a DVRP model based on the benchmark proposed by Kilby et al. [8] and Montemanni et al. [12], i.e., the DVRP model is a periodic reoptimization model. A DVRP model has a sequence of static VRP instances which contain all the customer requests known at each particular time, the periodic reoptimization needs to process the unserved customer requests.

In the DVRP model, all vehicles depart from the same depot to service customer requests. A vehicle is allowed to return to the depot when all customer requests have been serviced or the capacity of the vehicle has exceeded.

The DVRP has three parameters which are different from static VRP [1], which are: 1) the *available time* indicating that when the customer request appeared; 2) the *duration* for each customer request; 3) the *working day*, which determines the available time to service the customer requests.

For periodic reoptimization, the working day is divided into time slices. When new customer requests arrive during a time slice, these requests are delayed to the end of the time slice. Similar to static VRP, the objective function of DVRP is to minimize the total distance of all the routes. To better understand the DVRPs, Fig. 1 shows an example of DVRP. At time t_0 , the routes of the vehicles were planned based on the initially known customer requests. At time t_1 , when vehicles were executing their routes, some new customer requests arrived, then the reoptimization procedure was executed to process the remaining unserved customers. At time t_2 , when all the customers were serviced, the vehicles returned to the depot.

3 Proposed BSO-DVRP Algorithm

For static VRPs, population based methods and local search or neighborhood search have been widely used. In the DVRPs, a difficult problem is when and how to choose good strategies to optimize the current state (time slice). A good strategy can not only avoid local optima, but also make use of the solutions obtained from previous time slice. In this paper, we choose BSO [17] as a framework to determine when and what strategy to choose, and choose ALNS [15] and ACS [3] from extensive heuristics for VRPs reported in literature as two good strategies to choose from.

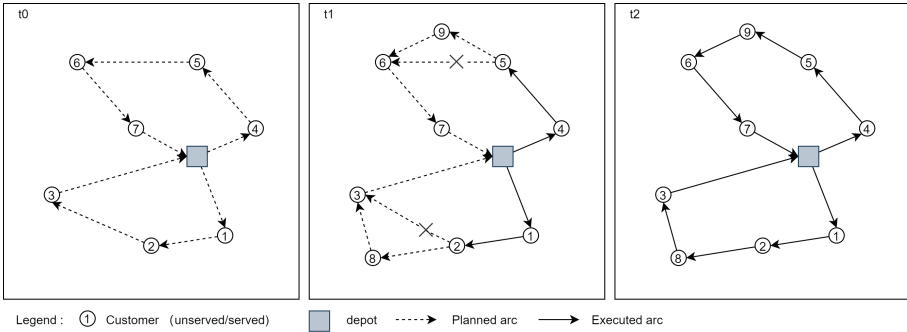


Fig. 1. An example of DVRP with

Algorithm 1. The BSO-DVRP Algorithm

- 1: initialize parameters and generate N solutions
 - 2: **while** not terminated **do**
 - 3: evaluate and sort N solutions according to the fitness function (total distance)
 - 4: take the top $perc_e$ percentage as elitists, and set the remaining as normals
 - 5: **if** $rand(0, 1) < p_{elitists}$ **then** \triangleright generate a new solution based on elitists
 - 6: **if** $rand(0, 1) < p_{one}$ **then**
 - 7: randomly select a solution S_i in elitists
 - 8: $S' \leftarrow ALNS(S_i)$
 - 9: **else**
 - 10: randomly select two solutions S_i, S_j in elitists
 - 11: $S' \leftarrow ACS(S_i, S_j)$
 - 12: **end if**
 - 13: **else** \triangleright generate a new solution based on normals
 - 14: **if** $rand(0, 1) < p_{one}$ **then**
 - 15: randomly select a solution S_i in normals
 - 16: $S' \leftarrow ALNS(S_i)$
 - 17: **else**
 - 18: randomly select two solutions S_i, S_j in normals
 - 19: $S' \leftarrow ACS(S_i, S_j)$
 - 20: **end if**
 - 21: **end if**
 - 22: **if** S' is better than S_i or S_i, S_j **then**
 - 23: replace S_i or the worse one between S_i and S_j
 - 24: **end if**
 - 25: **end while**
 - 26: **return** best solution in N solutions
-

ACS [3] is the most popular one among population based heuristics for VRP. The procedure of ACS is as the following: the ants depart from depot and select customers to service by pheromone one by one. When the ants can not service more customers, i.e., the vehicle capacity constraint can not be satisfied, the ants

will return to the depot. Then, the ants will restart from the depot to service remained customers. The pheromone is updated after the ants completed their tours.

ALNS [15] is the most popular algorithm among neighborhood search for VRP. There are some heuristics to remove customer requests and some other heuristics to insert removed requests into routes in ALNS. The adaptive method is used to select the removal heuristics and inserting heuristics according to the history information.

BSO [16, 18] is inspired by the human brainstorming process. In the BSO procedure, different strategies were applied for simulating the problem owners to choose good solutions that they believe in the brainstorming process. Since the clustering operation in the BSO algorithm is very time consuming, an enhanced version of BSO algorithm in objective space named BSO-OS [17] is proposed. In the BSO-OS algorithm, solutions are sorted according to their fitness values, and the top $perc_e$ percentage are set as elitists, while the remaining are set as normals. New solutions are generated in four ways: 1) generate a new solution based on one randomly selected elitist; 2) generate a new solution based on two randomly selected elitists; 3) generate a new solution based on one randomly selected normal; 4) generate a new solution based on two randomly selected normals.

To guide the choice of strategies, we applied the BSO-OS algorithm to choose different strategies, i.e., ALNS or ACS for the periodic reoptimization. In this paper, we propose a hybrid BSO algorithm named BSO-DVRP for solving the DVRPs. The pseudocode of the proposed algorithm is shown in Algorithm 1.

The overall frame of the algorithm is BSO-OS, in which ALNS and ACS are applied to generate new solutions. The new solutions are generated based on one solution or two solutions, either in elitists or in normals. If one solution S_i is selected, then ALNS is applied to perform a neighborhood search to generate a new solution. If two solutions S_i, S_j are selected, then these two solutions are sent to the ACS as the initial solution. By updating the pheromones, a new solution will be generated by ACS.

4 Experimental Results

4.1 Benchmark Description and Experiments Setup

In our experiments, we used the benchmarks proposed by Kilby et al. [8] and extended by Montemanni et al. [12], which were designed based on popular static VRP benchmarks proposed by Taillard [19], Christofides and Beasley [2], and Fisher [4]. There are three datasets in the benchmark: the first dataset has 12 instances with 75 to 150 customers; the second dataset consists of seven instances with 50 to 199 customers; and the third dataset contains two instances with 71 and 134 customers. In the DVRP instances, three types of data were introduced: 1) *available time* indicating that when the request first appeared; 2) *duration* for each request; 3) *working day*, which is the length of the day.

According to [13], when the number of time slices n_{ts} is set as 25, the trade-off between the total distance and the computational cost is well balanced. In addition, the working day is set as $16 \times 25 = 400$. The parameter settings of the BSO-OS algorithm are listed in Table 1.

Table 1. Parameter settings for BSO-OS

$perc_e$	$p_{elitists}$	p_{one}	max_iter
0.1	0.2	0.6	150

In Table 1, $perc_e$ is the percentage of elitists, $p_{elitists}$ is the probability of choosing elitists, p_{one} is the probability of choosing one solution, and max_iter is the maximum number of iterations.

The proposed algorithm was programmed in Python, and the DVRP experiments were conducted on an Intel Xeon E5-2650 CPU@2.30 GHz PC with 16 GB RAM.

4.2 Comparative Study

To evaluate the proposed algorithm, a comparative study has been conducted. The compared algorithms for DVRP are ACS [12], DVRP-GA and TS [6], DAPSO and VNS [7], and the GA-based DVRP [1]. The experimental results are given in Table 2, in which the best result for each instance is highlighted with bold fonts. The gap of the total distance (TD) is computed according to Eq. (1).

$$Gap = \frac{TD_{ours} - TD_{best}}{TD_{ours}} \quad (1)$$

where TD_{ours} is the total distance obtained by the proposed algorithm, and TD_{best} is the best total distance among all the compared algorithms for DVRP. A negative gap illustrates that the proposed algorithm outperforms all the other algorithms for DVRP, while a positive gap shows that the proposed algorithm is worse than the best result obtained by other algorithms.

As is observed from Table 2, 12 out of 21 new best solutions were found by the proposed algorithm, and the gap between our solutions and the previously best solutions ranges from -8.85% to -0.27% . From Table 2, the other nine solutions obtained by the proposed algorithm have a gap range from 0.11% to 6.91% , which are also competitive. All these results showed that the proposed algorithm is effective to generate high quality solutions for the DVRPs.

Table 2. Experimental results and comparison with literature for DVRP

Instance	ACS	DVRP-GA	TS	DAPSO	VNS	GA-based DVRP	BSO-DVRP	Gap (%)
c50	631.30	603.57	570.89	575.89	599.53	566.01	597.18	5.22
c75	1009.36	981.51	981.57	970.45	981.64	944.46	938.79	-0.60
c100	973.26	997.15	961.10	988.27	1022.92	943.89	912.83	-3.40
c100b	944.23	891.42	881.92	924.32	866.71	869.41	900.48	3.75
c120	1416.45	1331.80	1303.59	1276.88	1285.21	1288.66	1173.10	-8.85
c150	1345.73	1318.22	1348.88	1371.08	1334.73	1273.50	1254.47	-1.52
c199	1771.04	1750.09	1654.51	1640.40	1679.65	1646.36	1631.23	-0.56
tai75a	1843.08	1778.52	1782.91	1816.07	1806.81	1744.78	1713.46	-1.83
tai100a	2375.92	2208.85	2232.71	2249.84	2250.50	2181.31	2237.90	2.53
tai150a	3644.78	3488.02	3328.85	3400.33	3479.44	3280.79	3391.39	3.26
tai75b	1535.43	1461.37	1464.56	1447.39	1480.70	1441.35	1426.93	-1.01
tai100b	2283.97	2219.28	2147.70	2238.42	2169.10	2119.03	2068.12	-2.46
tai150b	3166.88	3109.23	2933.40	3013.99	2934.86	2885.94	2899.56	0.47
tai75c	1574.98	1406.27	1440.54	1481.35	1621.03	1433.73	1392.96	-0.96
tai100c	1562.30	1515.10	1541.28	1532.56	1490.58	1504.63	1479.48	-0.75
tai150c	2811.48	2666.28	2612.68	2714.34	2674.29	2593.78	2596.67	0.11
tai75d	1472.35	1430.83	1399.83	1414.28	1446.50	1408.48	1503.79	6.91
tai100d	2008.13	1881.91	1834.60	1955.06	1969.94	1793.64	1855.26	3.32
tai150d	3058.87	2950.83	2950.61	3025.43	2954.64	2911.47	3073.54	5.27
f71	311.18	280.23	301.79	279.52	304.32	288.30	270.28	-3.42
fl34	15135.51	15717.90	15528.81	15875.00	15680.05	14871.40	14831.29	-0.27

5 Conclusions

In this paper, we proposed a BSO algorithm named BSO-DVRP for solving DVRPs. By cutting the working day into a certain number of time slices, the DVRP was transformed into static VRP to solve. To guide the choice of strategies for the periodic reoptimization, the BSO-OS algorithm was applied. In addition, two popular heuristics, ALNS and ACS were selected for generating new solutions in the proposed BSO procedure.

The experiments were conducted on the DVRP benchmark, and a comparative study with six algorithms for DVRP was carried out. A total of 12 out of 21 new best solutions were found by the proposed algorithm, and the other solutions obtained by the proposed algorithm were also competitive, which illustrated the effectiveness of the proposed algorithm.

Acknowledgment. This work is supported by the National Science Foundation of China under the Grant No. 61761136008, the Shenzhen Peacock Plan under the Grant No. KQTD2016112514355531, the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under the Grant No. 2017ZT07X386, the Science and Technology Innovation Committee Foundation of Shenzhen under the Grant No. ZDSYS201703031748284.

References

1. AbdAllah, A.M.F., Essam, D.L., Sarker, R.A.: On solving periodic re-optimization dynamic vehicle routing problems. *Appl. Soft Comput.* **55**, 1–12 (2017)
2. Christofides, N., Beasley, J.E.: The period routing problem. *Networks* **14**(2), 237–256 (1984)
3. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**(1), 53–66 (1997)
4. Fisher, M.L., Jaikumar, R.: A generalized assignment heuristic for vehicle routing. *Networks* **11**(2), 109–124 (1981)
5. Golden, B.L., Raghavan, S., Wasil, E.A.: *The Vehicle Routing Problem: Latest Advances and New Challenges*, vol. 43. Springer, Heidelberg (2008)
6. Hanshar, F.T., Ombuki-Berman, B.M.: Dynamic vehicle routing using genetic algorithms. *Appl. Intell.* **27**(1), 89–99 (2007)
7. Khoudja, M.R., Sarasola, B., Alba, E., Jourdan, L., Talbi, E.G.: A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests. *Appl. Soft Comput.* **12**(4), 1426–1439 (2012)
8. Kilby, P., Prosser, P., Shaw, P.: *Dynamic VRPs: a study of scenarios*. University of Strathclyde Technical Report, pp. 1–11 (1998)
9. Larsen, A.: *The dynamic vehicle routing problem* (2000)
10. Lenstra, J.K., Kan, A.R.: Complexity of vehicle routing and scheduling problems. *Networks* **11**(2), 221–227 (1981)
11. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1998)
12. Montemanni, R., Gambardella, L.M., Rizzoli, A.E., Donati, A.V.: A new algorithm for a dynamic vehicle routing problem based on ant colony system. In: *Second International Workshop on Freight Transportation and Logistics*, vol. 1, pp. 27–30 (2003)
13. Montemanni, R., Gambardella, L.M., Rizzoli, A.E., Donati, A.V.: Ant colony system for a dynamic vehicle routing problem. *J. Comb. Optim.* **10**(4), 327–343 (2005)
14. Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. *Eur. J. Oper. Res.* **225**(1), 1–11 (2013)
15. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **40**(4), 455–472 (2006)
16. Shi, Y.: Brain storm optimization algorithm. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) *ICSI 2011*. LNCS, vol. 6728, pp. 303–309. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21515-5_36
17. Shi, Y.: Brain storm optimization algorithm in objective space. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1227–1234. IEEE (2015)
18. Shi, Y.: An optimization algorithm based on brainstorming process. In: *Emerging Research on Swarm Intelligence and Algorithm Optimization*, pp. 1–35. IGI Global (2015)
19. Taillard, É.: Parallel iterative search methods for vehicle routing problems. *Networks* **23**(8), 661–673 (1993)
20. Taillard, É.D., Gambardella, L.M., Gendreau, M., Potvin, J.Y.: Adaptive memory programming: a unified view of metaheuristics. *Eur. J. Oper. Res.* **135**(1), 1–16 (2001)