Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 175 (2020) 447–452

Procedia Computer Science

www.elsevier.com/locate/procedia

The 15th International Conference on Future Networks and Communications (FNC)
August 9-12, 2020, Leuven, Belgium

Search for optimal routes on roads applying metaheuristic algorithms

Amelec Viloria [a,*], Nelson Alberto Lizardo Zelaya[b], Noel Varela[c]

[a,c] *Universidad de la Costa, Barranquilla, Colombia.*
[b]*Universidad Tecnológica Centroamericana (UNITEC), San Pedro Sula, Honduras*

Abstract

The design of efficient routes for vehicles visiting a significant number of destinations is a critical factor for the competitiveness of many companies. The design of such routes is known as the vehicle routing problem. Indeed, efficient vehicle routing is one of the most studied problems in the areas of logistics and combinatorial optimization. The present study presents a memetic algorithm that evolves using a mechanism inspired by virus mutations. Additionally, the algorithm uses Taboo Search as an intensification mechanism.

Keywords: The problem of routing vehicles; Limited capacity; Memetic algorithm.

1. Introduction

The vehicle routing problem integrates the bin packing problem and the travelling salesman problem. The bin packing problem assigns destinations to each of the vehicles and the traveling salesman problem designs the routes for each vehicle [1].

Since vehicle routing is NP-Difficult, route designs are made using approximation algorithms called metaheuristics. The metaheuristic algorithms allow to find quality solutions in acceptable computing times. For NP-

* Corresponding author. Tel.: +57-3046238313.
E-mail address: aviloria7@cuc.edu.co

1877-0509
10.1016/j.procs.2020.07.063

Difficult problems, it is not feasible to determine if such solutions are optimal, except for some cases in which the value of the objective function of the solution found coincides with a lower/upper level. Some of the algorithms used in the literature to solve instances of the vehicle routing problem with limited capacity are: simulated annealing; taboo search [2]; ant colony [3] and genetic algorithms [4].

Memetic Algorithms have been successfully used to solve problems characterized as NP-Difficult. Memetic Algorithms. In [5], The authors combine algorithms inspired by principles of evolution, with other methods such as: Taboo Search and Simulated Annealing; such integration produces more versatile and efficient algorithms. For more information about Genetic Algorithms, the reader can consult [6].

The problem of vehicle routing studied in this study is the following: given a group of destinations to which goods must be delivered using a certain number of vehicles with limited capacity; designing the routes of the different vehicles minimizing the total distance traveled, ensuring that their capacity is not exceeded and visiting each destination only once.

2. Memetic Algorithm

Given the NP-difficult nature of the vehicle routing problem, approximation algorithms (metaheuristics) are required to find quality solutions in acceptable computing times. The metaheuristic algorithm presented in this section is a Memetic Algorithm, MEMVRP, which integrates an evolutionary algorithm inspired by the viral mutation with the principles of the Taboo Search presented in [7]. The following sections describe each of the components of MEMVRP.

2.1. Initial solution

The starting point of most heuristic algorithms is the generation of an initial solution. The algorithms in charge of generating the initial solutions are called constructor algorithms. The constructor algorithm used in this study, Mconst, assigns each target only once and, in a random way, among the different vehicles. For example, a possible initial solution is to assign the destinations 8, 7, 6, 4, 3 to vehicle 1 and the destinations 5, 2, 11, 10, 9 to the second vehicle to be visited in those sequences. It is important to remember that each route begins and ends at the base (i=1). The distance traveled by the first vehicle is 19(d1,8) + 34(d8,7) + 18(d7,6) + 33(d6,4) + 37(d4,3) +17(d3,1) = 168 km and the distance traveled by the second vehicle is 212 km, for a target value of 168+212 = 380 km. Note that each vehicle carries an initial load of 10 tons [8].

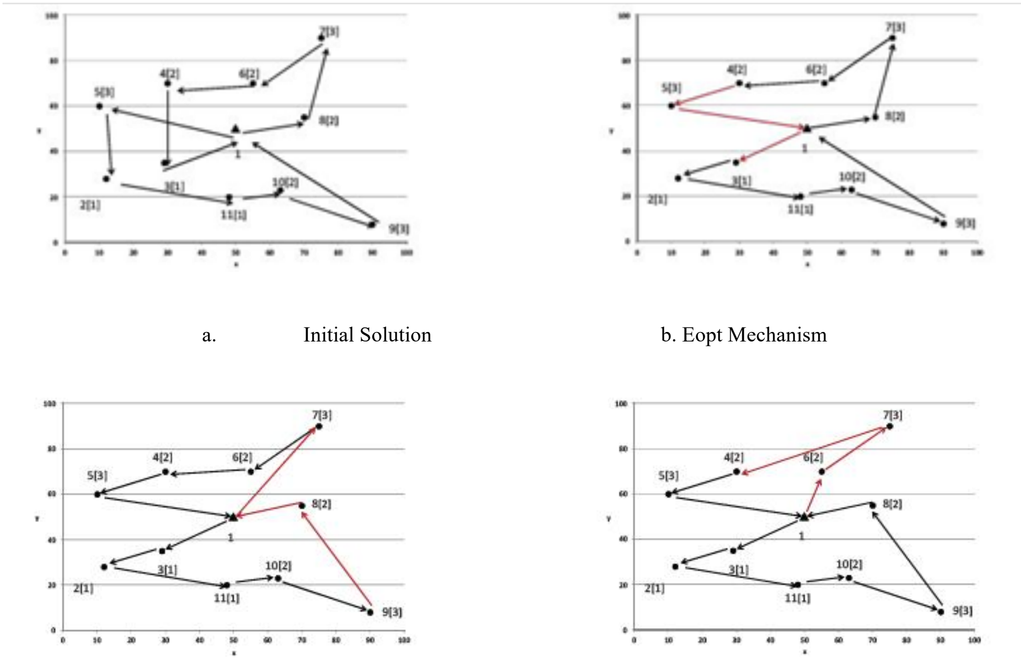
2.2. Representation of the solution

The solution obtained in section 2.1 is achieved by assigning the value of 1 to the variables x1.8, x8.7, x7.6, x6.4, x4.3, x3.1, x1.5, x5.2, x2.11, x11.10, x1.1, x10.9, x9.1 and 0 to all the others. Given the practical limitations of mathematical formulation, it is important to use a representation that facilitates the process of finding quality solutions using heuristic algorithms. In fact, each solution can be represented using a set of vectors So ={[s1, si,,… sj, s1]1, [s1, sn. ,…, sm, s1]2, …, [s1, so,,…, sp, s1]K}, where each vector represents a path that starts and ends at the base (s1). The solution obtained by Mconst can be represented as So = {[1, 8, 7, 6, 4, 3, 1]1 [1,5, 2, 11, 10, 9, 1]2}. Considering that all paths start and end at the base, the representation can be simplified to obtain S0 = {[8, 7, 6, 4, 3]1[5, 2, 11, 10, 9]2}.

2.3. Local search

The local search explores different solutions starting from a known solution. To obtain new solutions, the local search uses different mechanisms to alter the current solution. The mechanisms applied in this study are called Eopt and Iopt. These mechanisms alter existing routes by exchanging destinations (Eopt) or by removing and inserting destinations (Iopt). An example of the Eopt mechanism is to exchange destinations 5 and 3 in So to obtain the solution S1 = {[8, 7, 6, 4, 5]1 [3, 2, 11, 10, 9]2} with a target value of 357 km. It is important to note that the solution obtained is not viable, since the initial load of vehicle 1 is 12 tons [9]. Figure 2b illustrates S1. Often non-viable solutions are accepted at the initial stage of the search. In general terms, non-viable solutions can guide the search towards viable solutions of better quality. An example of the Iopt mechanism is to remove destination 8 from the first route and insert it at the end of the second route to obtain S2 = {[7, 6, 4, 5]1 [3, 2, 11, 10, 9, 8]2}. Figure 1b illustrates S1. Non-viable solutions are often accepted in the initial phase of the search.

The new solution is feasible and has a target function of 362 km (Figure 1c). The Eopt and Iopt mechanisms are not limited to exchanging destinations between different routes, but can also be applied to a single route. For example, exchanging destinations 6 and 7 on route 1 generates the solution S3 = {[6, 7, 4, 5]1 [3, 2, 11, 10, 9, 8]2} with a target function of 360 km. This last solution is optimal for the instance (Figure 1d).



a. Initial Solution b. Eopt Mechanism

c. Iopt mechanism d. Eopt mechanism

Fig. 1. Initial solution and search mechanisms

In general terms the local search can be described as follows [10][11]:

a. Generate an S0 starter solution using Mconst.
b. Apply the Eopt and Iopt mechanisms to S0. The set of solutions obtained in this process is called the solution neighborhood (V) and each exchange generated by the Eopt and Iopt mechanisms is called a move.
c. Select the best V solution (S*).

   i. If S* is better than S0, convert S* to S0 and repeat steps a, b and c. The execution of steps a, b and c is called iteration.
   ii. if S* is not better than S0, finish the algorithm. When this happens, the search is said to have reached a local optimum.

2.4. Taboo search

Taboo Search [12] is one of the most widely used metaheuristic algorithms in combinatorial optimization. Taboo Search uses a number of mechanisms to allow the search to escape from the optimal locations and continue towards better solutions. Taboo Search uses two types of memory: short-term memory and long-term memory. Short term memory (Taboo list) allows the search to proceed without repeating previously explored solutions.

The Taboo Search can be summarized in the following steps:

a. Generate an S0 starter solution using Mconst
b. Smejor = S0
c. Apply Eopt and Iopt a S0 thoroughly
d. If a taboo move improves Smejor, remove the move from the Taboo list
e. Generate V considering only non-taboo moves
f. Penalizing non-viable solutions
g. Select the best V solution (S*)
h. If S* is better than Smejor, Smejor = S*
i. So = S*. Note that the search does not stop if the quality of S* is lower than So
j. Repeat steps c until a predetermined number of iterations is not reached without exceeding Smejor
k. Finish the algorithm.

2.5. Memetic Algorithm

The greatest strength of the Taboo Search is the concentration on exploring promising areas of the solution space (intensification). On the other hand, the greatest weakness is the exploration of new regions of the solution space (diversification). On the other hand, algorithms inspired by evolutionary theories such as Genetic Algorithms [13] have efficient diversification mechanisms. The greatest limitation of Genetic Algorithms in vehicle routing is in the reproductive mechanism. Genetic Algorithms combine two solutions (parents) to generate a third one (child). Most of the solutions obtained by this method generate routes that visit the same destinations more than once. These routes are not viable and must be repaired before applying the Taboo Search. Figure 3 is an example of a typical case. The first two rows. P1 and P2, contain the parents' information. The third row, RA, is a random number between 0 and 1. The mechanism for obtaining P3 is to select the RA-based targets. If RA > 0.5, the destination of P1 is selected, otherwise the destination of P2 is selected.

The resulting routes, P3, visit destinations 5, 6 and 7 twice and do not include destinations 2, 8 and 10. Solutions like this are not part of the solution space. To avoid these difficulties, MEMVRP uses a mechanism inspired by the evolutionary processes of viruses, Mvirus, which generates multiple mutations to an existing solution. The mutations are based on multiple Eopts applied sequentially and chosen at random. This mechanism keeps the number of destinations associated with each vehicle constant and ensures that each destination is visited only once. Table 1 exemplifies the process for obtaining P1* from P1. Note that the solution obtained is not viable since the initial load of vehicle 2 is 11 tons, but it is part of the solution space.

Table 1. Mvirus Mechanism

	Vehicle 1					**Vehicle 2**				
P1	7	3	7	2	10	5	11	2	6	11
P1’	7	3	11	2	10	5	11	2	6	9
P1’’	10	3	10	2	10	4	7	2	6	9
P1*	**11**	**4**	**10**	**2**	**10**	**5**	**5**	**2**	**6**	**9**

MEMVRP can be summarized in the following steps [12]:

a. Create the initial population of solutions, MPob, using Mconst.
b. Apply TSmem to each solution in MPob.
c. Select the best solution and save it as Mbest.
d. Apply Mvirus and TSmem to each of the solutions in MPob.

e. Select the lowest quality solution in MPob and replace it with Mbest.
f. Select the lowest quality remaining solutions in MPob and:
g. Remove them from MPob.
h. Create new solutions using Mconst.
i. Apply TSmem to each of the new solutions
j. Add the new solutions to MPob.
k. Select the best solution in MPob and compare it with Mbest. If it is of better quality, replace Mbest.
l. Repeat steps d through g until the desired number of generations is completed. Each repetition creates a new generation of solutions.

Step f replaces the mutation mechanisms used in Genetic Algorithms. The application of Taboo Search makes these mechanisms ineffective. The mechanism used in f is based on the one proposed in Mendes, et AL. 2005.

3. Results

MEMVRP was evaluated using a recognized set of instances for vehicle routing called "Augerat set A". This set is one of the most used to evaluate algorithms for vehicle routing and has the advantage that its optimal solutions are known. It is important to mention that obtaining such solutions requires thousands of hours of computational time on advanced equipment. The Memetic Algorithm was coded in Visual C++ 2010 Express Edition. In the evaluation of the algorithm a computer was used with Intel Core i5 2.4 GHz processor with 8 GB of memory and Windows 7 64 bits [14].

MEMVRP parameters were tuned based on an experiment design that considered three levels for each of the parameters. Three replicates were made and the combination that generated the best results was chosen. Three replicates were made to find the best combination:

a. Population: populations of 5, 10 and 15 solutions were considered. The best results were found for the level of 10 solutions. It seems that considering fewer solutions generates a convergence around low quality solutions. More solutions do not generate quality improvements and increase computational time appreciably.
b. Generations: 3 levels were considered, 20, 30 and 40 generations. The best solutions were found with 40 generations. A lower number of generations is insufficient for the algorithm to arrive at quality solutions. A higher number increases the computational time significantly without improving the quality of the solution.
c. Replacements. Three levels 0, 2 and 4 replacements were considered. The best responses were found for the level that replaces the 2 lowest quality solutions in each generation with new solutions.
d. Stay on the Taboo list: four levels were considered, 0.15*(number of destinations), 0.30*(number of destinations) and 0.45*(number of destinations). The best results were found for values of 0.30*(number of destinations). Apparently, very low values allow the generation of cycles, while higher values affect the intensification process.
e. Penalty: three levels were considered.1.0/ (number of destinations), 1.5/ (number of destinations) and 2.0/ (number of destinations). The increase in the factor for penalizing inefficient solutions is 1.5/ (number of destinations) per iteration. Lower values generate non-viable solutions, while higher values restrict the solution space, affecting the quality of the final solution.
f. Completion criteria: 10*(number of destinations), 15*(number of destinations) and 20*(number of destinations) levels were considered. Levels above 10*(number of destinations) did not improve the quality of the solutions. Lower values end the intensification process very soon, generating solutions of inferior quality.

When comparing MEMVRP with the Taboo Search presented in [15], TS, it can be seen that MEMVRP performed better. Indeed, MEMVRP obtained the optimal solution for all instances evaluated while TS failed in problems A-n60-k9 and A-n61-k9. Additionally, when comparing the number of times that each algorithm was able to find the optimal solution in 10 attempts, it can be seen that MEMVRP equals or exceeds TS in all cases except for A-n39-k6. It is also possible to observe the deterioration in the quality of the solutions obtained by TS for problems with 60 or more destinations. On the other hand, MEMVRP maintains a much more stable performance, finding the optimal solution in 99.5% of the attempts. Finally, it is important to mention that MEMVRP is robust with respect to the values of the different parameters, which facilitates its implementation.

4. Conclusions

This paper presents a Memetic Algorithm called MEMVRP to solve the problem of vehicle routing with limited capacity. ME-MVRP uses a mechanism inspired by virus mutation to create new generations of solutions. Additionally, MEMVRP makes use of Taboo Search to improve each of the new generation solutions. The performance of MEMVRP is superior to the Taboo Search presented in [16].

References

[1] Peng, B., Zhang, Y., Lü, Z., Cheng, T. C. E., & Glover, F. (2020). A learning-based memetic algorithm for the multiple vehicle pickup and delivery problem with LIFO loading. Computers & Industrial Engineering, 142, 106241.
[2] Yağmur, E., & Kesen, S. E. (2020). A memetic algorithm for joint production and distribution scheduling with due dates. Computers & Industrial Engineering, 142, 106342.
[3] Li, X., Shi, X., Zhao, Y., Liang, H., & Dong, Y. (2020). SVND Enhanced Metaheuristic for Plug-In Hybrid Electric Vehicle Routing Problem. Applied Sciences, 10(2), 441.
[4] Wan, J., Chen, X., & Li, R. (2020). Improved Memetic Algorithm for Multi-depot Multi-objective Capacitated Arc Routing Problem. In MATEC Web of Conferences (Vol. 308, p. 01002). EDP Sciences.
[5] Mor, A., & Speranza, M. G. (2020). Vehicle routing problems over time: a survey. 4OR, 1-21.
[6] Nosrati, M., & Khamseh, A. (2020). Bi objective hybrid vehicle routing problem with alternative paths and reliability. Decision Science Letters, 9(2), 145-162.
[7] Deng, J., & Wang, L. (2017). A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem. Swarm and evolutionary computation, 32, 121-131.
[8] Duran-Micco, J., Vermeir, E., & Vansteenwegen, P. (2020). Considering emissions in the transit network design and frequency setting problem with a heterogeneous fleet. European Journal of Operational Research, 282(2), 580-592.
[9] Neira, D. A., Aguayo, M. M., & Klapp, M. A. (2020). New compact integer programming formulations for the multi-trip vehicle routing problem with time windows. Computers & Industrial Engineering, 106399.
[10] Smiti, N., Dhiaf, M. M., Jarboui, B., & Hanafi, S. (2020). Skewed general variable neighborhood search for the cumulative capacitated vehicle routing problem. International Transactions in Operational Research, 27(1), 651-664.
[11] M'Hallah, R. (2020). Combining Exact Methods to Construct Effective Hybrid Approaches to Vehicle Routing. In Women in Industrial and Systems Engineering (pp. 435-456). Springer, Cham.
[12] Blocho, M. (2020). Parallel algorithms for solving rich vehicle routing problems. In Smart Delivery Systems (pp. 185-201). Elsevier.
[13] Molina, J. C., Salmeron, J. L., & Eguia, I. (2020). An ACS-based Memetic algorithm for the Heterogeneous Vehicle Routing Problem with time windows. Expert Systems with Applications, 113379.
[14] Ghosh, M., Begum, S., Sarkar, R., Chakraborty, D., & Maulik, U. (2019). Recursive memetic algorithm for gene selection in microarray data. Expert Systems with Applications, 116, 172-185.
[15] Viloria, A., Guerrero, I. M., Caraballo, H. M., Llinas, N. O., Valero, L., Palma, H. H., … Lezama, O. B. P. (2019). Effect on the demand and stock returns: Cross-sectional of big data and time-series analysis. In Communications in Computer and Information Science (Vol. 1122 CCIS, pp. 211–220). Springer. https://doi.org/10.1007/978-981-15-1301-5_17
[16] Viloria, A., Crissien Borrero, T., Vargas Villa, J., Torres, M., García Guiliany, J., Vargas Mercado, C., … Batista Zea, K. (2019). Differential evolution clustering and data mining for determining learning routes in moodle. In Communications in Computer and Information Science (Vol. 1071, pp. 170–178). Springer Verlag. https://doi.org/10.1007/978-981-32-9563-6_18