IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 3, September 2024, pp. 2683~2693
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i3.pp2683-2693

The cooperative algorithm with auxiliary objectives for the truck and trailer routing problem

Ricardo Pérez-Rodríguez[1], Luis Eduardo Urbán-Rivero[2]
[1]Department of Applied Mathematics, Faculty of Engineering, CONAHCYT-Universidad Autónoma de Querétaro, Querétaro, México
[2]Department of Systems, Faculty of Sciences and Engineering, Universidad Autónoma Metropolitana, Azcapotzalco Campus, México City, México

Article Info

Article history:

Received Sep 28, 2023
Revised Oct 25, 2023
Accepted Jan 15, 2024

Keywords:

Auxiliary objectives
Combinatorial optimization
Cooperative algorithm
Evolutionary algorithms
Heuristics
Multi-objectivization
Vehicle routing

ABSTRACT

In this paper, a cooperative algorithm with auxiliary objectives is proposed to resolve the truck and trailer routing problem. In this proposal, each member of the population does not represent a complete solution as in almost any evolutionary algorithm. In addition, for each member, an aptitude is not possible to compute based only on its codification, because the member has only partial information of the solution. All the members of the population have partial information of the solution. Therefore, these members need to cooperate to obtain an aptitude for the entire population. This way of computing fitness is clearly a gap in the literature, and must be investigated. Moreover, the multi-objectivization approach incorporates an important feature to the proposed algorithm in order to improve its performance, i.e., the multi-objectivization approach permits to identify the best trips using the auxiliary objectives. Enough experimental results are shown that the cooperative algorithm is competitive against other current evolutionary algorithms. There no exist statistically significant difference between the cooperative algorithm and the others.

Corresponding Author:

Ricardo Pérez-Rodríguez
Department of Applied Mathematics, Faculty of Engineering, CONAHCYT-Universidad Autónoma de Querétaro
Querétaro 76010, México
Email: dr.ricardo.perez.rodriguez@gmail.com

1. INTRODUCTION

In a classical truck and trailer routing problem (TTRP), the goal is to deliver merchandises to customers by trucks and trailers. We consider only one depot where the all the vehicles depart to different destinations, and these vehicles return to the same depot. Not only capacity constraints are included in the TTRP, also operational constraints are included, basically narrow spaces for maneuvers, traffic restrictions, among others. Therefore, in the TTRP, if there exist aforementioned restrictions in a customer location, then the vehicle must be parked at another place, unhitch the trailer, and continue the trip using only the truck, before arriving to that customer with limited access for the trailer. After attend the customer, the truck can continue attending other customers or returns to the trailer, and hitch it to continue the trip. This previous situation produces mainly three types of routes, i.e., routes using only trucks, routes using truck and trailer for those customers that permit full access without maneuvering restrictions, and routes using truck and trailer for those customers with maneuvering restrictions. However, if on the route there exist customers with maneuvering restrictions, the trailer must be unhitched before arriving to those customers. Let name to those customers that only permit access with the truck as "truck customers", to those customers that permit access with truck and trailer as "vehicle customers", to those routes using only truck as "truck routes", to those routes using truck and trailer as “vehicle routes”, and to those routes using truck and trailer with maneuvering restrictions as “mix routes”. Figure 1 details an example on it, i.e., the type of customers and the type of routes.

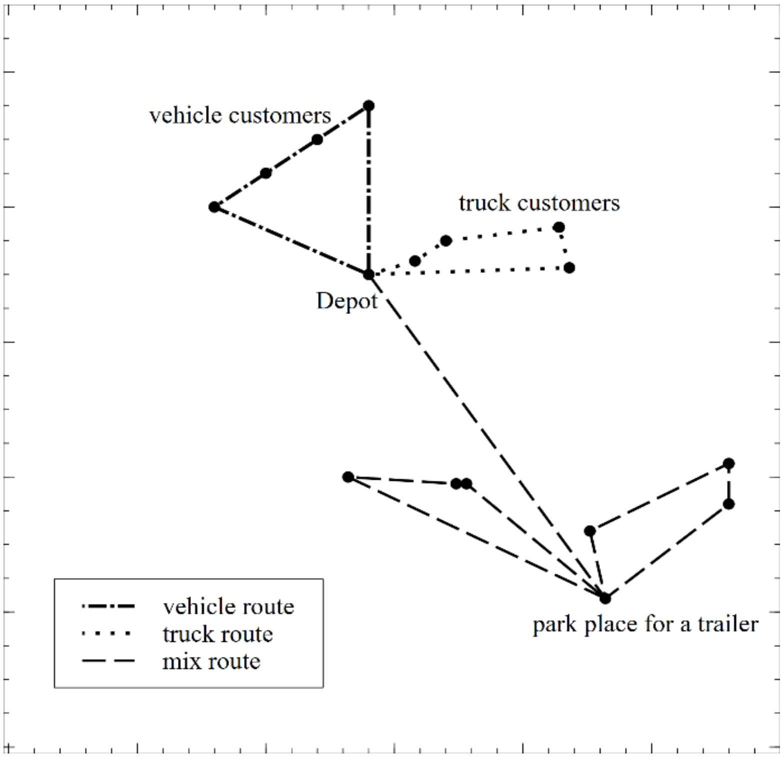

Figure 1. Type of routes in the TTRP

Based on the characteristics of the TTRP, it is suitable to apply evolutionary algorithms to find the best solution. The purpose is to build many routes as possible, and select the best set of routes of minimum total distance. It can be seen as a single-objective.

Currently the majority of methods to resolve vehicle routing problems use a conventional mechanism to build solutions, i.e., group customers in a route, and then sequence the route. It is commonly named ‘cluster-first route-second’. The constraints of the problem being analyzed are considered to group customers. Prins *et al.* [1] cited contributions of this approach for vehicle routing optimization problems, such as [2]–[4] cited important contributions for the TTRP using this approach. Examples of applications in real-world situations are found in [5]–[11]. However, for two decades, an alternative approach has had increasing acceptance, i.e., the ‘route-first cluster-second’ mechanism. This relatively new approach has led to successful methods for routing problems. It is due to its flexibility and efficiency. Such properties have let to resolve the TTRP too. Some researchers [1], [4] cited the most relevant papers in this category, such as [12]–[19]. More updated contributions about general routing problems can be found in [20]–[24].

In the route-first cluster-second, generally each solution for the TTRP, is represented for a permutation of vertices. Therefore, after the split-phase, a fitness is obtained for that solution. In this research, also we adopt the route-first cluster-second approach, but we differ in the split-phase to build routes. In addition, it is accepted that the evolutionary algorithms normally require some type of method that helps to enhance their performance. Therefore, we address the gap combining the multi-objectivization method to improve the results.

One of the methods to enhance the performance of the evolutionary algorithms for optimizing single-objectives is the multi-objectivization. It consists of increasing the efficiency of an evolutionary algorithm by using auxiliary objectives [25]–[27]. Being more specific, the methods from Pareto-based multi-objective optimization may be helpful when we solve optimization problems with a single-objective [28], [29]. However, relatively there have been very few works about it. Some contributions can be found in [30]–[32]. Basically, the multi-objectivization decomposes a single objective into several auxiliary objectives [33].

These aforementioned main ideas were used in order to enhance the performance of the cooperative algorithm (CoopA). In the CoopA, unlike other evolutionary algorithms, the members of the population do not compete among them in order to survive. Instead, the members participate and cooperate to obtain a population fitness. In the CoopA, each member has no fitness of his own. On the contrary, the population fitness belongs to everyone. Therefore, in the CoopA does make sense to select members to be crossed and mutated. Instead, the competition is between generations, i.e., all the offspring against all the parents. However, as any evolutionary algorithm, the CoopA has drawbacks such as lack of diversity of the solutions and poor ability of exploitation. In this research, a hybridization strategy is considered to address the issue. Fundamentally, the hybridization of the multi-objectivization and the CoopA is detailed.

In each generation, thousands of solution vectors, i.e., members of a population, are randomly generated. After, each member is decomposed to produce two auxiliary objectives by the [34] approach. The Pareto-based multi-objective optimization method (NSGA-II), proposed by [35], is used to identify the first Pareto-front using the aforementioned auxiliary objectives. Then, the selected members of the first Pareto-front are added to those members generated by heuristics in order to create a combined population. In this research, the resulting population is used to produce feasible routes to resolve the TTRP. At the beginning, the CoopA does not consider the original objective, i.e., the total travel distance for the TTRP. Instead, each member randomly generated is used to produce two auxiliary objectives. In the CoopA, each member from the initial population is a permutation-based representation, as in almost any vehicle routing representation.

According to Jähne *et al.* [34], each member of the population is randomly divided into two sets. Each set is the complement of the other. By two equations, provided by [34], we compute the fitness for each set. After, the NSGA-II finds the first Pareto-front from the sets previously computed, and the selected members of the aforementioned process are added to those members generated by heuristics. Finally, it originates the cooperative algorithm with auxiliary objectives (CoopAwAO). Base of the results shown in the results and comparison section, the hybridization permits to enhance the performance of the CoopA. Also it permits to tackle its main drawbacks, i.e., lack of diversity of the solutions and poor ability of exploitation.

2. PROBLEM STATEMENT

Previously we discuss the possible routes that can be built in the TTRP, i.e., truck routes, vehicle routes, and mix routes. Also, erstwhile we refer to truck customers, and vehicle customers. Then we can establish binary parameters for each type of route. It means:

$$a_{ij} \begin{cases} 1 \text{ if the customer } i \text{ is visited in the truck route } j \\ 0 \text{ otherwise} \end{cases}$$

$$b_{ik} \begin{cases} 1 \text{ if the customer } i \text{ is visited in the vehicle route } k \\ 0 \text{ otherwise} \end{cases}$$

$$c_{im} \begin{cases} 1 \text{ if the customer } i \text{ is visited in the mix route } m \\ 0 \text{ otherwise} \end{cases}$$

In addition, we can define binary variables:

$$x_{j} \begin{cases} 1 \text{ if the route } j \text{ is selected and used in the solution for the TTRP} \\ 0 \text{ otherwise} \end{cases}$$

$$y_{k} \begin{cases} 1 \text{ if the route } k \text{ is selected and used in the solution for the TTRP} \\ 0 \text{ otherwise} \end{cases}$$

$$z_{m} \begin{cases} 1 \text{ if the route } m \text{ is selected and used in the solution for the TTRP} \\ 0 \text{ otherwise} \end{cases}$$

The objective function considers to minimize the total distance of the solution as a single-objective, where:

$$d_j = \text{represents the total distance of the route } j$$

$$d_k = \text{represents the total distance of the route } k$$

$$d_m = \text{represents the total distance of the route } m$$

The set of feasible truck routes is named *J*, the set of feasible vehicle routes is named *K*, and the set of feasible mix routes is named *M*. The set of truck customers is named N_t, and the set of vehicle customers is named N_v. Then we have:

$$\min z = \sum_{j\in J} d_j x_j + \sum_{k\in K} d_k y_k + \sum_{m\in M} d_m z_m \quad (1)$$

subject to:

$$\sum_{j\in J} a_{ij} x_j + \sum_{k\in K} b_{ik} y_k + \sum_{m\in M} c_{im} z_m = 1 \quad \forall i \in N_v \quad (2)$$

$$\sum_{j\in J} a_{ij} x_j + \sum_{m\in M} c_{im} z_m = 1 \quad \forall i \in N_t \quad (3)$$

$$x_j \in \{0,1\}, \quad \forall j \in J \quad (4)$$

$$y_k \in \{0,1\}, \quad \forall j \in K \quad (5)$$

$$z_m \in \{0,1\}, \quad \forall j \in M \quad (6)$$

The objective function (1) consists of the first part that corresponds to the total distance of truck routes, the second part represents the total distance of vehicle routes, and the third part is the total distance of mix routes. Constraints (2) assure that each vehicle customer is visited exactly once; whereas, constraints (3) assure that each truck customer is visited exactly once by a truck route or by a mix route.

3. THE COOPAWAO FRAMEWORK

3.1. Permutation-based representation: Route-first step

Each member of the population is a permutation-based representation. A permutation representation is built to execute the route-first step. Each element represents a customer to visit in the TTRP. We set a thousand of members of size n, where n indicates the number of customers to visit in a TTRP instance.

3.2. Splitting each member of the population into two sets: The multi-objectivization phase

As in others vehicle routing problems, a set of n customers $c_1, \ldots, c_n$ and an associated $n\ x\ n$ distance matrix M can be defined. The entries in M represent the distances between the customers, so $M(c_1, c_2)$ is the distance from c_1 to c_2, where $M(c_1, c_2) = M(c_2, c_1)$. If $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a permutation of $(1, 2, \ldots, n)$ representing the tour of the customers. In this step, we compute the distance associated with the tour as:

$$D(\pi) = \sum_{i=1}^{n} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}) \quad where\ i \oplus 1 = \begin{cases} i+1 & if\ i < n \\ i & if\ i = n \end{cases} \quad (7)$$

After, we use auxiliary objectives, i.e., S_1 and S_2, to compute the fitness for each set. It is using the equations from Jähne *et al's*. [34] approach. Let:

$$S_1(\pi, p) = \sum_{i=1}^{|p|} M\left(c_{\pi[\pi^{-1}[p[i]]\ominus 1]}, c_{p[i]}\right) + M(c_{p[i]}, c_{\pi[\pi^{-1}[p[i]]\oplus 1]})$$

$$S_2(\pi, q) = \sum_{i=1}^{|q|} M\left(c_{\pi[\pi^{-1}[q[i]]\ominus 1]}, c_{q[i]}\right) + M(c_{q[i]}, c_{\pi[\pi^{-1}[q[i]]\oplus 1]}) \quad (8)$$

Where p is a subset of $\{1, 2, \ldots, n\}$, q is the complementary set of p, and $\ominus 1$ is the reverse of $\oplus 1$. The two new objectives $S_1(\pi, p)$ and $S_2(\pi, q)$ are the sum of distances in the path incident on the customers in p and q, respectively.

3.3. The Pareto-based multi-objective optimization phase

The NSGA-II is implemented to obtain the first Pareto-front by fitness of all the members. The corresponding members are selected to proceed to the next stage. NSGA-II is an evolutionary algorithm developed as an answer to the shortcomings of early evolutionary algorithms, which lacked elitism and used a sharing parameter in order to sustain a diverse Pareto set. NSGA-II uses a fast non-dominated sorting algorithm, sharing, elitism, and crowded comparison. Elitism implies that the best solutions of the previous iteration are kept unchanged in the current one. This significantly increases the convergence speed of the algorithm. Additionally, its use of a fast non-dominated sorting algorithm contributes to a significant reduction of its computational complexity. For more details, the interested reader is referred to [35].

3.4. Heuristics

The CoopAwAO uses different heuristics to build permutation representations too, i.e., trips. All the procedures are well-known techniques in the literature. A heuristic or heuristic technique, is any approach to problem solving or self-discovery that employs a practical method that is not guaranteed to be optimal, perfect, or rational, but is nevertheless sufficient for reaching an immediate, short-term goal or approximation in a search space.

3.4.1. Random insertion

The first one, it is a random procedure, where all the vertices are positioned on the trip randomly. The trip is filled from left to right with each vertex randomly selected. The trip is complete when all the vertices already are set in the trip in some position.

3.4.2. Nearest neighbor technique

The second, the nearest neighbor procedure, where it starts at a random vertex and repeatedly visits the nearest vertex until all the vertices have been visited. Again, the trip is filled from left to right. The first position is randomly filled with some vertex. Next, we need to compute the distance between the previous chosen vertex and the rest of the vertices. The nearest vertex is chosen to put on the next position on the trip.

3.4.3. Nearest neighbor technique from both end-points

The third, the nearest neighbor procedure from both end-points, where it starts with a vertex chosen randomly. Then, it continues with the nearest unvisited vertex to this vertex. We will have two end vertices. We add a vertex to the trip such that this vertex has not visited before and it is the nearest vertex to these two end vertices. We update the end vertices. It ends after visiting all the vertices.

3.4.4. Nearest insertion technique

The fourth, the nearest insertion procedure, where it begins with two vertices. It then repeatedly finds the vertex not already in the trip that is closest to any vertex in the trip, and places it between whichever two vertices would cause the resulting trip to be the shortest possible. It stops when no more insertions remain.

3.4.5. 2-Opt technique

Finally, the fifth, the 2-opt procedure proposed by [36], where it originates from the idea that trips with edges that cross over are not optimal. 2-opt will consider every possible 2-edge swap, swapping 2 edges when it results in an improved trip. The 2-opt algorithm works as follows: take 2 arcs from the route, reconnect these arcs with each other and calculate new travel distance. If this modification has led to a shorter total travel distance the current route is updated. The algorithm continues to build on the improved route and repeats the steps.

3.5. Combined population

The members obtained from the step 3.3., and 3.4., are combined into a single population. It means that a combined population is created adding selected members from the Pareto-front, obtained from NSGA-II procedure, and members from heuristics, previously detailed. All the members have the same length, i.e., all of them are trips. Each trip contains n vertices.

3.6. Building feasible routes: cluster-second step

Each trip, from the combined population, is split as many feasible routes as possible. For that purpose, three variants are used to build feasible routes. The first variant creates feasibles routes using only trucks as main vehicles. The second variant creates feasibles routes using the truck and trailer but visiting only vehicle customers. The third variant creates feasibles routes using the truck and trailer, and it visits either truck customer or vehicle customer as a mix route.

3.6.1. Truck routes construction

In this case, we read each trip from left to right. Let a trip $T = \{v_i, \dots, v_j, \dots, v_k\}$, we confirm the expression $q_i \leq Q_t$, it means that if the demand of the vertex i (q_i) is less or equal to the capacity truck Q_t, then the vertex i can belong to the route. Otherwise, the route is finished. The process continues reading the trip, and we update the total demand on this route if the next vertex j can be considered on the route, i.e., if the vertex j meets $Q_{ij} \leq Q_t$ where $Q_{ij} = \sum_{u=i}^{j} q_{v_u}$. We will stop when such condition is not met, then the route is finished. The process continues reading the rest of the trip, and it finishes when all the vertices have already been assigned to some route.

3.6.2. Vehicle routes construction

The process is very similar than the previous one. The main difference is found in the capacity of the vehicle, which is no longer Q_t, will be $Q_t + Q_r$, i.e., the capacity of the truck plus the capacity of the trailer. In addition, we need to verify if the vertex i can receive a trailer. Otherwise, the route is finished. The process continues reading the trip, and we update the total demand on this route if the next vertex j meets the restriction of capacity and reception of a trailer. We will stop when such conditions are not met, then the route is finished. The process continues reading the rest of the trip, and it finishes when the vertices able to receive a trailer have already been assigned to some route.

3.6.3. Mix routes construction

The process starts reading the trip as the previous ones. The capacity of the vehicle is $Q_t + Q_r$. We confirm the expression $q_i \leq Q_t + Q_r$, then the vertex i can belong to the route. Otherwise, the route is finished. The process continues reading the trip, and we update the total demand on this route if the next vertex j can be considered on the route, i.e., if the vertex j meets $Q_{ij} \leq Q_t + Q_r$ where $Q_{ij} = \sum_{u=i}^{j} q_{v_u}$. We will stop when such condition is not met, then the route is finished.

The next step is to verify if the route, already built, contains at least one truck customer. If so then, we confirm that the first vertex on the route be a vehicle customer. If so then, the route is a mix route, and we park and unhitch the trailer is that first vertex. If not then, the route is unfeasible and it is discarded. The process continues reading the rest of the trip, and it finishes when we have already analyzed all the vertices on the trip. All the routes built by these three variants are now members of the population, in the CoopAwAO framework. All the routes are considered to find a fitness for the population.

3.7. Optimization

3.7.1. Total distance computing

For each route built by any of the three aforementioned variants, a total distance is computed. The total distance for the truck routes and the vehicle routes is easily calculated because it corresponds to a single tour, without forgetting that the route leaves the depot and returns at the end. The total distance for the mix routes is calculated considering that the trailer is unhitch at the first vehicle customer location, after that the truck visits one or more customers on the route, probably the truck has to come back to the parking place of the trailer to transfer product between the trailer and the truck, and continue the tour until satisfying pending customers. We emphasize that the route leaves the depot, sometime the truck has to return to hitch the trailer, and finally the vehicle goes back to the depot at the end.

3.7.2. Fitness of the population

In the crudest terms, fitness involves the ability of organisms or, more rarely, populations or species to survive and reproduce in the environment in which they find themselves. The consequence of this survival and reproduction is that organisms contribute genes to the next generation. The mathematical model, detailed in section 2, is applied to minimize the total distance of the solution, i.e., the fitness of the population. This model considers all the routes built in section 3.6, and the total distance of each route computed in section 3.7.1, to identify the minimum, and know which routes are elected.

3.8. Offspring

Again, we create trips by different procedures. Four of them, have been previously detailed in section 3.6., i.e., the nearest neighbor technique, the nearest neighbor technique from both end-points, the nearest insertion technique, and the 2-Opt technique. The fifth procedure is the partially mapped crossover, called partially mapped crossover (PMX) genetic operator. Here, we select randomly two trips, obtained in section 3.5., and we apply the PMX operator to produce one new trip. The process detailed in 3.6., is repeated to produce feasible routes that we consider as the offspring in the CoopAwAO framework. The process 3.7., is repeated to know the fitness of the offspring.

3.9. Replacement

The population with the best fitness survives, the other is eliminated. Although the population with the best fitness survives, the best trips of both populations are preserved to build feasible routes in the next generation. The CoopAwAO framework is shown as follows:

Pseudocode CoopAwAO framework
$D_0 \leftarrow$ Generate M trips
$D_1 \leftarrow$ Select the best N trips from multi − objectivization

D_2 ← Generate M − N trips by heuristics
D_3 ← Combine $D_1 \cup D_2$
R_0 ← Build feasible routes from D_3
DistR_0 ← For each route, compute distance from R_0
Best ← Find the fitness from R_0, and DistR_0
BestR ← Store the routes as the best
BestDist ← Store the distances as the best
t ≔ 1
Do
    S_t ← Generate M trips
    S_1 ← Select the best N trips from multi − objectivization
    S_2 ← Generate M − N trips by heuristics
    S_3 ← Combine $S_1 \cup S_2$
    R_t ← Build feasible routes from S_3
    DistR_t ← For each route, compute distance from R_t
    BestOffspring ← Find the fitness from R_t, and DistR_t
    Best ← if apply, update the best solution from BestOffspring
    BestR ← if apply, replacement BestR from R_t
    BestDist ← if apply, replacement BestDist from DistR_t
    t ≔ t + 1
Until (stopping criterion is met)
Output: Best

4. RESULTS AND DISCUSION

The CoopAwAO performance is compared with other evolutionary algorithms. The comparison is done using the algorithm detailed by [37], the simulated annealing heuristic designed by [38], and the bat algorithm presented by [39]. All these algorithms were implemented following the available information.The set of instances used in the comparison is found at [40]. Figure 2 details the performance for each algorithm.

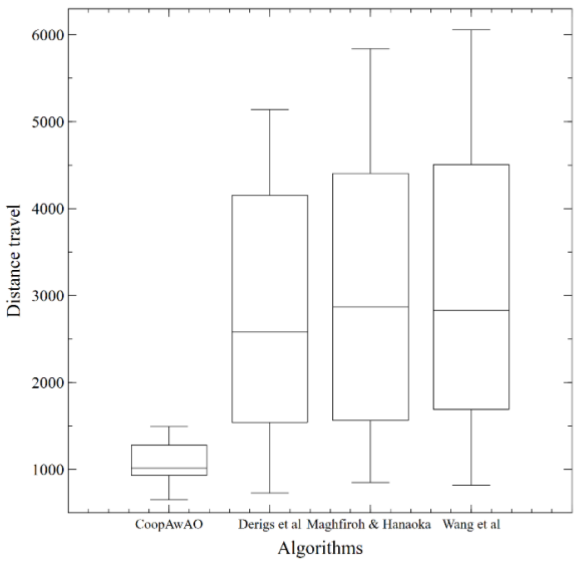

Figure 2. First computational results

Based on Figure 2, the dispersion of the results is less in the CoopAwAO than others. It is due to the replacement procedure, detailed in section 3.6., it keeps the best fitness over all the iterations, and the average of each generation cannot be far from the best solution because the most offspring are built by the same procedures than the parents. In addition, the multi-objectivization approach permits to identify the best trips using the auxiliary objectives in order to improve the performance of the CoopAwAO. In addition, another comparison is presented in Figure 3. It is using the algorithm proposed by [41], and the procedure shown by

[42] for comparison with the CoopAwAO scheme. The results of these algorithms were taken directly from the available literature.

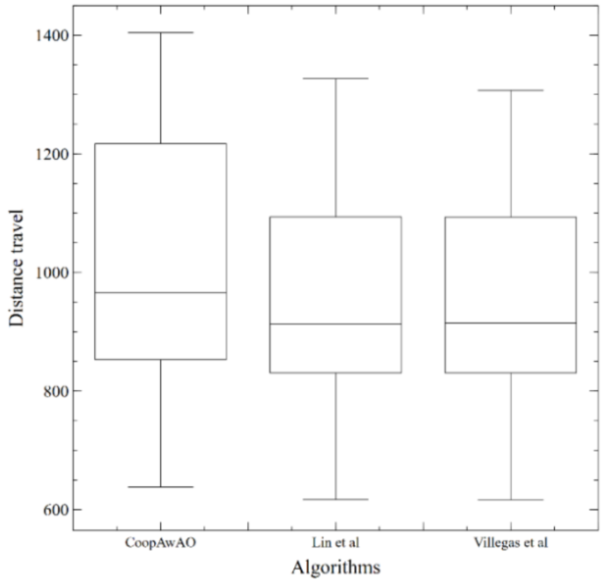

Figure 3. Second computational results

Based on Figure 3, the dispersion of the results is very similar among the algorithms. The performance of the CoopAwAO is competitive. It is due to the large number of routes built in each instance. We devised procedures, detailed in section 3.6., to tackle the most drawback of the set-partitioning model for the TTRP, i.e., the structure of mix routes that normally are resolved by column generation and branch-and-price methods [43]. Furthermore, in this research, we do not use any auxiliary graph to build feasible routes.

A Dunnett test is done to identify if there exist statistically significant difference between the CoopAwAO, and the other methods. The CoopAwAO is competitive, there no exist statistically significant difference as shown in Figure 4. Villegas *et al.* [4] indicated that the set-partitioning model for the TTRP is often impractical. It is due to the huge number of feasible routes, and since it is impossible to compute all of them, the CoopAwAO scheme builds a considerable number of them to tackle the aforementioned drawback. Table 1 details the number of routes computed by the CoopAwAO scheme, and the best solution founded for the instance number one.

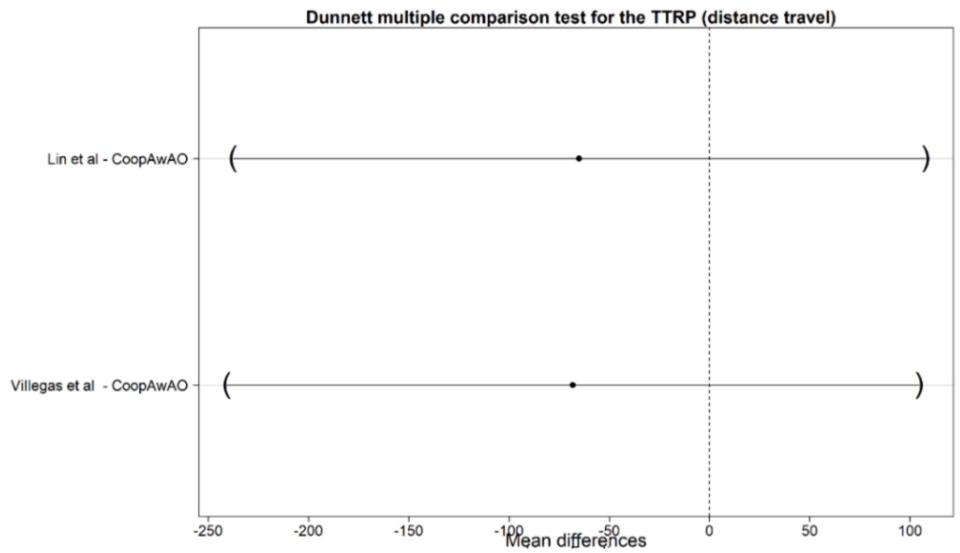

Figure 4. Dunnett test

Table 1. Number of feasible routes and the best solution founded

ID instance	Number of routes	Best found solution
1	1,396	625.853
	4,197	592.273
	7,056	581.493
	9,807	579.903
	12,560	573.213
	14,015	566.056

5. CONCLUSIONS

The CoopAwAO scheme is suitable to tackle the TTRP. It is a well-known nondeterministic polynomial time-hard (NP-hard) issue. The main drawback of the set-partitioning model, i.e., the inability to compute all the routes, is cleverly resolved by devised procedures. The CoopAwAO scheme is competitive. It was not necessary to incorporate auxiliary graphs to create feasible routes for those possible mix routes. The set of instances used in the comparison are considered benchmarking. Therefore, the use of the Dunnett test is clearly justified and forceful. The performance of the CoopAwAO scheme should be taken into account in the literature. The proposal of the CoopAwAO, i.e., considers all the members of the population to obtain a fitness for the all the population is substantial. Each member participates and cooperates to identify the fitness of the population. It is obtained by choosing from the routes of minimum total distance. Although each member of the CoopAwAO scheme only has partial information of the solution for the population, it is not a drawback for the CoopAwAO scheme, on the contrary, this enriches it is performance by consider many routes in the solution as s shown in Table 1. The multi-objectivization incorporates an important feature to the CoopAwAO in order to improve its performance, i.e., the multi-objectivization approach permits to identify the best trips using the auxiliary objectives. As future work, other greedy procedures should be implemented to create better trips, to help the CoopAwAO to find more suitable routes. In addition, other procedures should be incorporated to get offspring, to enhance the performance of the CoopAwAO, not only the Pareto-front approach and/or auxiliary objectives. Other optimization problems should be resolved by the CoopAwAO scheme, in order to confirm its performance.

REFERENCES

[1] C. Prins, P. Lacomme, and C. Prodhon, "Order-first split-second methods for vehicle routing problems: a review," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 179–200, Mar. 2014, doi: 10.1016/j.trc.2014.01.011.

[2] F. Semet, "A two-phase algorithm for the partial accessibility constrained vehicle routing problem," *Annals of Operations Research*, vol. 61, no. 1, pp. 45–65, Dec. 1995, doi: 10.1007/BF02098281.

[3] J. C. Gerdessen, "Vehicle routing problem with trailers," *European Journal of Operational Research*, vol. 93, no. 1, pp. 135–147, Aug. 1996, doi: 10.1016/0377-2217(95)00175-1.

[4] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "A matheuristic for the truck and trailer routing problem," *European Journal of Operational Research*, vol. 230, no. 2, pp. 231–244, Oct. 2013, doi: 10.1016/j.ejor.2013.04.026.

[5] A. D. Pia and C. Filippi, "A variable neighborhood descent algorithm for a real waste collection problem with mobile depots," *International Transactions in Operational Research*, vol. 13, no. 2, pp. 125–141, Mar. 2006, doi: 10.1111/j.1475-3995.2006.00539.x.

[6] A. M. F. -Fard, M. R. -Bourani, N. Cheikhrouhou, and M. H. -Keshteli, "Novel modifications of social engineering optimizer to solve a truck scheduling problem in a cross-docking system," *Computers & Industrial Engineering*, vol. 137, Nov. 2019, doi: 10.1016/j.cie.2019.106103.

[7] A. Hoff and A. Løkketangen, "A tabu search approach for milk collection in western Norway using trucks and trailers," *Arbeidsnotat*, pp. 1–30, 2008.

[8] L. Levy and L. Bodin, "Scheduling of local delivery carriers for the United States postal service," in *Arc Routing: Theory, Solutions and Applications*, Boston, MA: Springer US, 2000, pp. 419–442.

[9] O. Theophilus, M. A. Dulebenets, J. Pasha, Y. Lau, A. M. F. -Fard, and A. Mazaheri, "Truck scheduling optimization at a cold-chain cross-docking terminal with product perishability considerations," *Computers & Industrial Engineering*, vol. 156, Jun. 2021, doi: 10.1016/j.cie.2021.107240.

[10] R. Vahrenkamp, *Transportation logistic in rural setting: the case of milk collection*, Department of Business and Economics, University of Kassel, Kassel, Jerman, 1989.

[11] V. Yu, T. Lin, and C.-C. Lu, "An ant colony system for solving the truck and trailer routing problem," *Journal of the Chinese Institute of Transportation*, vol. 23, no. 2, pp. 199–218, 2011.

[12] M. Boudia, C. Prins, and M. Reghioui, "An effective memetic algorithm with population management for the split-delivery vehicle routing problem," in *Hybrid Metaheuristics*, vol. 4771, pp. 16–30, 2007, doi: 10.1007/978-3-540-75514-2_2.

[13] N. Labadi, C. Prins, and M. Reghioui, "A memetic algorithm for the vehicle routing problem with time windows," *RAIRO - Operations Research*, vol. 42, no. 3, pp. 415–431, Jul. 2008, doi: 10.1051/ro:2008021.

[14] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002, Oct. 2004, doi: 10.1016/S0305-0548(03)00158-8.

[15] J. Renaud, F. F. Boctor, and G. Laporte, "An improved petal heuristic for the vehicle routeing problem," *Journal of the Operational Research Society*, vol. 47, no. 2, pp. 329–336, Feb. 1996, doi: 10.1057/jors.1996.29.

[16] D. M. Ryan, C. Hjorring, and F. Glover, "Extensions of the petal method for the vehicle routing," *Journal of the Operational Research Society*, vol. 44, no. 3, pp. 289–296, Mar. 1993, doi: 10.1057/jors.1993.54.

[17] J. G. Villegas, C. Prins, and C. Prodhon, "GRASP/evolutionary local search hybrids for a truck and trailer routing problem," in *MIC 2009: The VIII Metaheuristics International Conference*, pp. 1-10, 2009.

[18] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 780–794, Aug. 2010, doi: 10.1016/j.engappai.2010.01.013.

[19] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "GRASP/VND with path relinking for the truck and trailer routing problem," *TRISTAN VII (7th triennial symposium on transportation analysis). Tromsø, Norway*, pp. 789–792, 2010.

[20] V.F. Yu, S.-W. Lin, L. Zhou, and R. Baldacci, "A fast simulated annealing heuristic for the multi-depot two-echelon vehicle routing problem with delivery options" *Transportation Letters*, 2023, 1–12, doi: 10.1080/19427867.2023.2257923.

[21] S. Wang, X. Zhu, P. Shang, X. Lin, L. Yang, and L. Tavasszy, "Two-echelon multi-commodity multimodal vehicle routing problem considering user heterogeneity in city logistics," *Expert Systems with Applications*, vol. 252, 2024, doi: 10.1016/j.eswa.2024.124141.

[22] A. Fragkogios, Y. Qiu, G. K. D. Saharidis, and P. M. Pardalos, "An accelerated benders decomposition algorithm for the solution of the multi-trip time-dependent vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 317, no. 2, pp. 500-514, 2024, doi: 10.1016/j.ejor.2024.04.013.

[23] J. Lehmann and M. Winkenbach, "A matheuristic for the two-echelon multi-trip vehicle routing problem with mixed pickup and delivery demand and time windows," *Transportation Research Part C: Emerging Technologies*, vol. 160, 2024, doi: 10.1016/j.trc.2024.104522

[24] A. Polimeni, A. Vitetta, "Network design and vehicle routing problems in road transport systems: Integrating models and algorithms," *Transportation Engineering*, vol. 16, Jun. 2024, doi: 10.1016/j.treng.2024.100247.

[25] F. Neumann and I. Wegener, "Can single-objective optimization profit from multiobjective optimization?" in *Natural Computing Series*, Springer, 2008, pp. 115–130.

[26] F. Neumann and I. Wegener, "Minimum spanning trees made easier via multi-objective optimization," *Natural Computing*, vol. 5, no. 3, pp. 305–319, Sep. 2006, doi: 10.1007/s11047-006-9004-x.

[27] T. Brys, A. Harutyunyan, P. Vrancx, M. E. Taylor, D. Kudenko, and A. Nowe, "Multi-objectivization of reinforcement learning problems by reward shaping," in *2014 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2014, pp. 2315–2322, doi: 10.1109/IJCNN.2014.6889732.

[28] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler, "Multiobjective genetic programming: reducing bloat using SPEA2," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001, vol. 1, pp. 536–543, doi: 10.1109/CEC.2001.934438.

[29] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization (EMO'01)*, 2001, vol. 1993, pp. 269–283, doi: 10.1007/3-540-44719-9_19.

[30] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, "Do additional objectives make a problem harder?," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, Jul. 2007, pp. 765–772, doi: 10.1145/1276958.1277114.

[31] S. Watanabe and K. Sakakibara, "Multi-objective approaches in a single-objective optimization environment," in *2005 IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1714–1721, 2005, doi: 10.1109/CEC.2005.1554895.

[32] Z. Yan, L. Zhang, L. Kang, and G. Lin, "A new MOEA for multi-objective TSP and its convergence property analysis," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2003, pp. 342–354, doi: 10.1007/3-540-36970-8_24.

[33] I. Petrova and A. Buzdalova, "Selection of auxiliary objectives in the travelling salesman problem using reinforcement learning," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, Jul. 2015, pp. 1455–1456, doi: 10.1145/2739482.2764646.

[34] M. Jähne, X. Li, and J. Branke, "Evolutionary algorithms and multi-objectivization for the travelling salesman problem," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, Jul. 2009, pp. 595–602, doi: 10.1145/1569901.1569984.

[35] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[36] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, no. 6, pp. 791–812, Dec. 1958, doi: 10.1287/opre.6.6.791.

[37] U. Derigs, M. Pullmann, and U. Vogel, "Truck and trailer routing—problems, heuristics and computational experience," *Computers & Operations Research*, vol. 40, no. 2, pp. 536–546, Feb. 2013, doi: 10.1016/j.cor.2012.08.007.

[38] M. F. N. Maghfiroh and S. Hanaoka, "Dynamic truck and trailer routing problem for last mile distribution in disaster response," *Journal of Humanitarian Logistics and Supply Chain Management*, vol. 8, no. 2, pp. 252–278, Aug. 2018, doi: 10.1108/JHLSCM-10-2017-0050.

[39] C. Wang, S. Zhou, Y. Gao, and C. Liu, "A self-adaptive bat algorithm for the truck and trailer routing problem," *Engineering Computations*, vol. 35, no. 1, pp. 108–135, Mar. 2018, doi: 10.1108/EC-11-2016-0408.

[40] "Truck and Trailer Routing Problem (TTRP)," *Global Logistics and Supply Chain Management Research Group.* Accessed: January 1, 2023. [Online]. Available: http://140.118.201.170/ttrp/

[41] S.-W. Lin, V. F. Yu, and S.-Y. Chou, “Solving the truck and trailer routing problem based on a simulated annealing heuristic,” *Computers & Operations Research*, vol. 36, no. 5, pp. 1683–1692, May 2009, doi: 10.1016/j.cor.2008.04.005.
[42] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, “A GRASP with evolutionary path relinking for the truck and trailer routing problem,” *Computers & Operations Research*, vol. 38, no. 9, pp. 1319–1334, Sep. 2011, doi: 10.1016/j.cor.2010.11.011.
[43] M. Drexl, “Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem,” *Revista de Metodos Cuantitativos para la Economia y la Empresa*, vol. 12, no. 1, pp. 5–38, 2011.

BIOGRAPHIES OF AUTHORS

Prof. Dr. Ricardo Pérez-Rodríguez got the Ph.D. Science and Technology, Industrial and Manufacturing Engineering. He is currently works as a senior lecturer and currently member of the National System of Researchers. He does research in evolutionary algorithms, artificial intelligence, and simulation optimization. He has written diverse papers published in journals, he has leaded thesis of students, and he has participated in different congresses, and he has written and published chapters and books. He can be contacted at email: dr.ricardo.perez.rodriguez@gmail.com.

Dr. Luis Eduardo Urbán-Rivero received a Bachelor’s degree in Computer Engineering at Universidad Autónoma Metropolitana (UAM), Azcapotzalco campus, in 2011 and a Master’s and Ph.D. in Optimization from the same university in 2014 and 2018 respectively. He had a postdoctoral fellowship at Center for Mathematics Research (CIMAT) Aguascalientes campus, from 2019 to 2021. He is currently an associated full-time professor at UAM. He is a member of the Mexican Society of Operations Research (SMIO) and the Mexican System of Researchers (CONAHCYT-SNI). He can be contacted at email: leur@azc.uam.mx.