**INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING**
***Int. J. Optim. Civil Eng., 2020; 11(1):75-99***

# SOLVING THE VEHICLE ROUTING PROBLEM BY A HYBRID IMPROVED PARTICLE SWARM OPTIMIZATION

M. Yousefikhoshbakht[*, †]
*Department of Mathematics, Faculty of Sciences, Bu-Ali Sina University, Hamedan, Iran*

## ABSTRACT

The capacity vehicle routing problem (CVRP) is one of the most famous issues in combinatorial optimization that has been considered so far, and has attracted the attention of many scientists and researchers today. Therefore, many exact, heuristic and meta-heuristic methods have been presented in recent decades to solve it. In this paper, due to the weaknesses in the particle swarm optimization (PSO), a hybrid-modified version of this algorithm called PPSO is presented to solve the CVRP problem. In order to evaluate the efficiency of the algorithm, 14 standard examples from 50 to 199 customers of the existing literature were considered and the results were compared with other meta-heuristic algorithms. The results show that the proposed algorithm is competitive with other meta-heuristic algorithms. Besides, this algorithm obtained very close answers to the best known solutions (BKSs) for most of the examples, so that the seven BKSs were produced by PPSO.

**Keywords:** capacity vehicle routing problem; particle swarm optimization; combined optimization problems; local search algorithms.

Received: 10 December 2020; Accepted: 8 February 2021

## 1. INTRODUCTION

Increasing population and expansion of urbanity has caused problems that are tied to daily life, and solving them is very important. Logistics and supply chain is one of these issues that has a key role in industry, and services today and includes issues that start from the beginning of the production of a good, and continue until the delivery of the relevant goods to the customer. In this cycle, the final price of the good is very vital because the reasonable price is one of the most important factors for the sale of a commodity, and makes

[*]Corresponding author: Department of Mathematics, Faculty of Sciences, Bu-Ali Sina University, Hamedan, Iran
[†]E-mail address: khoshbakht@basu.ac.ir (M. Yousefikhoshbakht)

it more profitable for the producer if the goods have high quality. In other words, the lower the final cost of a product, the more competitive the company is in selling goods and providing better services to the customer. Therefore, reducing the cost is one of the most important factors for earning more profits.

One way to reduce the cost of a commodity is to minimize the cost of transportation, so that it can be transferred from place to place with the least cost. Therefore, today, the importance of transportation issues is not hidden to anyone, and its real applications in daily life have led researchers to give more importance to it day by day, and examine it from different aspects. The capacity vehicle routing problem (CVRP) is one of the most widely used issues in transportation issues. In this problem, there are a number of customers in a geographical area with a warehouse [1-3], so that each customer needs a certain amount of goods that must be delivered to them by a fixed fleet of vehicles (Fig. 1). The purpose is to determine a set of nets that begin from the warehouse and end with it at the end, provided that:

- Each customer is visited by a vehicle exactly once.
- The total demand of customers of each tour does not exceed the capacity of the vehicle, which is considered Q.
- The least total cost for all tours of vehicles is obtained.

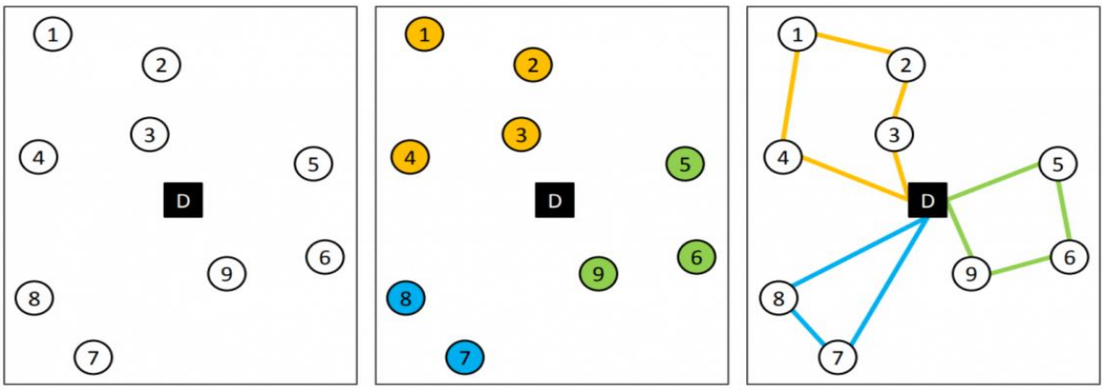


Figure 1. The steps of finding a solution of CVRP

On the other hand, the emergence of new real-world problems caused other restrictions to be imposed on the problem, and other versions such as the open vehicle routing problem (vehicles do not return to the warehouse after their mission and finish their route in the same final nodes) [4] (Fig. 2), the vehicle routing problem with pickup and delivery (each customer can request delivery and receive the goods, and the request is satisfied only once by only one vehicle) [5], the vehicle routing problem with time windows (each customer must be provided at a service interval) [6], the heterogeneous fixed fleet open vehicle routing problem (vehicles can be not the same as each one differs in capacity) [7], and the balanced vehicle routing problem (The distance traveled by vehicles should be as uniform as possible) [8].

Figure 2. The difference of the VRP (left) and OVRP solutions (right)

The CVRP issue and its spreads are one of the most important and widely used NP-hard problems that are widely used in the real optimization issues. Since the vehicle routing problem (VRP) was first introduced by Dantzig et al. in 1959 [9], many of its modes have been derived based on their different applications in the real world. In this paper, they presented a mathematical formulation and algorithm for solving this problem. Also, in their problem, the capacity of vehicles was equal to infinite and the stopping time in nodes was considered zero. The aim is to determine the routes for vehicles that each of them should meet a group of customers in a geographical area. Therefore, this is the primary issue of multiple traveling salesman problem (MTSP) in which there is no limit to visiting a customer.

The simplest version of the CVRP issue is the traveling salesman problem (TSP) because in this issue it is assumed that there is a vehicle with infinite capacity that must survey a number of nodes, and finally return to the starting place of movement. Although the TSP problem is just a very simple state of VRP issues, the process and results of the investigation on this issue directly affect the CVRP issues. It should be noted that to solve this problem, exact methods including branch and bound and dynamic planning were proposed. These methods were presented and upgraded until the theory of computational complexity was proposed. This theory pointed out that TSP is a difficult problem and there is no exact method that can reach the optimal solution in general at an acceptable time. Therefore, the methods of solving this problem changed, and moved toward heuristic methods. By proposing heuristic methods for solving the TSP problem, the way to develop heuristic methods for solving the VRP problems was also opened and such algorithms were used. For example, these algorithms include the method of generalized assignment heuristic by Fisher and J Kumar to solve the CVRP problem [10], which was efficient and able to solve large-scale problems. Besides, in these years, other heuristic methods have been presented to solve this problem. For example, a model presented by Christofides et al. [11] had excellent results and was able to solve a problem with 25 nodes and achieve the optimal solution. It should be noted that the formulation of such problems is done in two way, either in the form of probable planning of the integer and the mixed integer, or it uses Marcov's decision-making processes.

In [12], the vehicle routing problem with general soft time window is considered and a hybrid column generation is proposed to solve the problem. Each vehicle starts at the depot and should visit a number of geographically dispersed customers such that each one is visited only once during a time window. At first in this study, a mathematical model is developed and a hybrid column generation-metaheuristic algorithm is used to solve the problem efficiently. In more details, a quantum-inspired evolutionary method is applied for solving the sub-problems and the results of the column generation and electromagnetism are utilized to find high quality solutions. Finally, some modified standard problems are used to test the efficiency of the proposed algorithm.

In article [13], one of the applications of multi-attribute vehicle routing problem is presented to distribute the milk industry in the real world. In this problem, a heterogeneous fleet has been utilized to distribute goods that start from several warehouses and have resource constraints. Then, in order to obtain high-quality answers to these issues, a branch and price method is used. It should be noted that in the proposed method based on the problem structure, different strategies for branching are presented and compared. The results for different problems show that the proposed algorithm has a very good quality, and can achieve high quality solutions.

One version of the VRP problems that needs to be addressed today, due to its many applications, is the transportation of vital or expensive materials such as fuel shipments, vehicles carrying money, prisoner transport vehicles that may be robbed because of its importance. In article [14], this version of VRP is considered for the first time and a two-level model is presented for it. In addition, to solve the efficiency of this new problem in small sizes, a benders decomposition algorithm is presented and two valid inequalities have been used to increase the efficiency of the desired algorithm. A two-level meta-heuristic algorithm was also presented for large-size problems. Finally, the efficiency of these methods was tested for a number of random problems.

The important points about heuristic methods are that they do not have complexity, and get the answer of the problem in a short time. Although repeating the algorithm does not get different answers, these algorithms do not have an effective way to escape from local optimal points, and are often fall in local optimizations points. For this reason, scientists were looking for algorithms that would be more efficient than these methods and achieve the high quality solutions at an adequate time. These efforts led to the presentation of meta-heuristic methods. In these methods where the performance of the algorithm, unlike heuristic methods, depends on the user's decision, the CPU time of answers are obtained almost more than heuristic algorithms and less than exact methods [15-17]. Although the solutions of these methods are usually better than the answers of heuristic methods, and such algorithms use strategies that move as far as possible towards the global optimal.

One of the best modified local search methods used so far is tabu search, which still maintains its effectiveness despite the provision of new meta-heuristic methods. Osman presented methods based on simulated annealing and tabu search to solve the problem, which used efficiency strategies to search for problem space [18]. In addition, a new structure was used to use problem data, which reduced the CPU time to solve these problems by 50%. An efficient estimate was obtained for the size of the tabu list using the statistically tests, which has led to an increase in the efficiency of the algorithm in escaping from the optimal local points and obtained excellent results for standard problems. On the other hand, in addition to the 17 standard examples used, 9 random examples were considered. The results and comparisons show the efficiency of the proposed algorithms compared to other existing algorithms.

A hybrid algorithm including improved PSO and simulated annealing for several versions of VRP is proposed in [19]. This algorithm also used several effective strategies for improving the algorithm in order to increase ability for intensification and diversification mechanisms. Also, a number of removal and insertion techniques are applied in the PSO algorithm for better searching the feasible space. For evaluating of the algorithm, several instances of VRP, Open VRP, VRP with time windows and VRP with time windows and simultaneous delivery and pickup are considered and its results are compared with other algorithms. The computational experiments shows that this algorithm achieved excellent solutions and some best known solutions are improved.

In article [20], the problem of VRP with heterogeneous fleet is considered so that the delivery time of customers' orders as well as limited storage capacity of vehicles is very important in it. In order for the fleet to be encouraged to deliver goods to customers on a regular basis at certain times, there is a fine delay based on the type of goods and order. In addition, a mixed integer model has been presented for this problem to minims the cost of warehousing, transportation, delay and post-order cost as much as possible. Since this problem is an extension of the VRP issue, it belongs to the NP-hard problems, so exact algorithms such as the methods used in the CPLEX do not have the necessary efficiency to solve the large size problems. For this reason, the scatter search algorithm is presented to solve this problem, which could not be better efficiency to the iterated local search algorithm and the exact algorithm for these problems. Therefore, the proposed algorithm can be considered for the real problems of this version of the problem and produce high-quality solutions.

A new expansion of the OVRP issue is intended in [21] which several warehouses have been used to load goods to deliver to customers instead of a warehouse. Considering that today, manufacturing centers try to use rental fleets to deliver their products, which are usually produced in different regions, solving this problem is very practical and is needed in everyday issues of industry and services. For this reason, this problem has been investigated and a modified iterated local search algorithm has been proposed for it that has been able to produce high-quality answers compared to other metaheuristic algorithms. In this method, by counting each customer's movements and banning these movements, it causes the areas that have already been examined not to be considered again so that the algorithm can be guided to better areas. This correction has also caused the algorithm to escape from local optimal solutions as much as possible and achieve excellent answers.

The particle swarm optimization (PSO) is one of the most important meta-heuristic algorithms that has achieved excellent results in continuous problems. Despite the outstanding results of this algorithm compared to the meta-heuristic algorithms, it sometimes undergoes premature convergence and cannot maintain its efficiency during the implementation of the algorithm and escape from local optimization points. Therefore, to solve the CVRP problem, a modified combined PSO algorithm is presented in this paper, which has been able to achieve better results than other algorithms. In order to increase the efficiency of the proposed algorithm, in addition to several efficiency local search algorithms used to improve the answers, a well-organized strategy is used to move particles toward high-quality particles, which increases the power of the algorithm for intensification and diversification mechanisms. It should be noted that these two modifications cause a balance for global search and local search simultaneously in the algorithm and the problem space is examined efficiently.

In this paper, and in Section 2, the PSO method is first examined and then the modifications made on this algorithm are described in detail. In Section 3, the parameter setting are presented, and in the next section, the results of the proposed algorithm and other meta-heuristic algorithms are analyzed on the standard instances. Finally, in Section 5, the conclusions are presented.

## 2. SOLUTION METHOD

Swarm intelligence is a systematic property in which factors cooperate locally and the collective behavior of all factors causes a convergence at a point close to the optimal global solution, the strength of these algorithms is their lack of need for a nationwide control. Each particle in these algorithms has relative autonomy that can move across the answer space and must cooperate with other particles (factors). Two well-known algorithms are swarm intelligence, ant colony optimization and particle swarm optimization [22-30]. Both of these algorithms can be used to teach neural networks. In this section, the classical PSO algorithm is described, and the proposed modified algorithm called PPSO is presented.

### *2.1 The classic PSO*

The PSO algorithm is a collective search algorithm modeled on the social behavior of birds categories. At first, the algorithm was used to discover patterns governing the birds' flight at the same time and to suddenly change their tying and to change the optimal shape of the batch. In PSO, particles flow in the search space. The change of particle location in the search space is influenced by the experience and knowledge of themselves and their neighbors, so the other position of the particle mass affects how a particle is searched. The result of modeling this social behavior is the search process by which particles are directed toward successful regions. Particles learn from each other and go to their best neighbors based on the knowledge gained, the basis of the PSO's work is based on the principle that at any given moment, every particle adjusts its location in the search space according to the best place ever located and the best place in its entire neighborhood.

In 1995, Eberhart and Kennedy first proposed PSO as a meta-heuristic search method for optimization problems, inspired by the collective movement of birds looking for [31]. A group of birds search for food randomly in an atmosphere. There is only one piece of food in the search space. Each solution, called a particle, is the PSO in the algorithm equivalent to a bird in the birds' collective motion algorithm. Each particle has a value calculated by an objective function. The closer the particle in the search space to the target (the food in the bird movement model), the more competent it is. Also, each particle has a speed that directs the movement of the particle. Each particle continues to move in the problem space by following the optimal particles in the current state. At the first step of the algorithm, a group of particles are randomly generated, and try to find an optimal solution by updating generations. In each step, every particle is updated using the two best values. The first is the best situation that the particle has ever managed to achieve. The situation is known and maintained, which is also the best amount of nostalgia for that particle, which we display with the *pbest*. The best other value used by the algorithm is the global best position ever achieved by the particle population, which we call the *gbest* (Fig. 3).

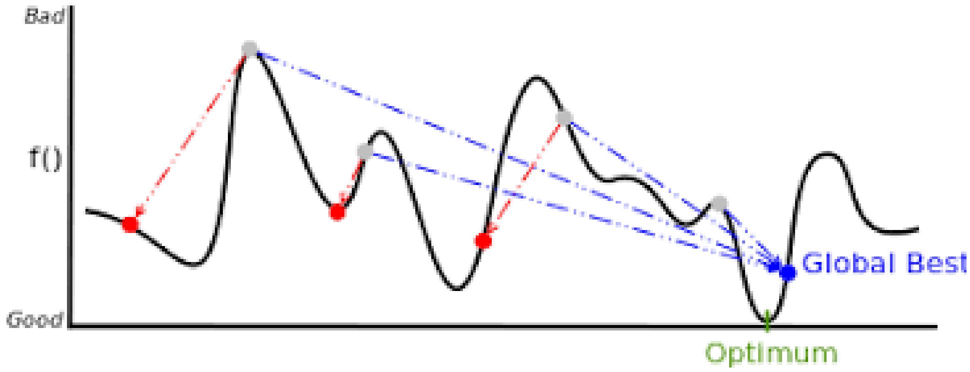


Figure 3. The movement of a particle to the *pbest* and *gbest* solutions

This process continues until the algorithm stop is satisfied. It should be noted that the condition of stopping in this algorithm is to rate birds to zero or reach the number of repetitions considered. In addition, each bird uses Formula (1) to determine the next position according to the values of *pbest* and *gbest* (Fig. 4). In this formula, $c_1$ and $c_2$ are constant learning parameters that determine the effect for *gbest* and *pbest*, and $r_1$ and $r_2$ are random numbers in the range [0,1]. Also, $x_{id}(t)$ shows the present position of each bird and $v_{id}(t)$ is the bird's movement in *t* stage. The best present position of any bird is shown by $p_{id}(t)$ and the best of $p_{id}(t)$ is presented by $g_{id}(t)$. Finally, w controls the iteration coefficient of birds' movement in the algorithm. It should be noted that the speed of movement is high at the beginning of the algorithm, but it will decrease during the CPU time of the algorithm.

$$x_{id}(t+1)=x_{id}(t)+w\,v_{id}(t)+c_1r_1\left(p_{id}(t)-x_{id}(t)\right)+c_2r_2\left(g_{id}(t)-x_{id}(t)\right) \qquad (1)$$

The relationship of (1) composed of two parts: the first part is the current position of the particle shown by $x_{id}(t)$, and the second part $v_{id}(t+1)=w\,v_{id}(t)+c_1r_1\left(p_{id}(t)-x_{id}(t)\right)+c_2r_2\left(g_{id}(t)-x_{id}(t)\right)$ shows speed of the particle at iteration *t+1*. This part includes speed of the particle at iteration *t*, and the movements toward to $p_{id}(t)$ and $g_{id}(t)$, respectively. If the third part in Formula (1) is not considered, the next particle's position is determined only according to the global search, else if the fourth part is eliminated, only the best particle experience $p_{id}(t)$ is considered, and the local search is attracted. Of course, this was just the beginning and extensive research was conducted to improve this method, which led to stronger versions of this method [32-34] that the reader could see a summary of the development, improvement and applications of this algorithm in [35].

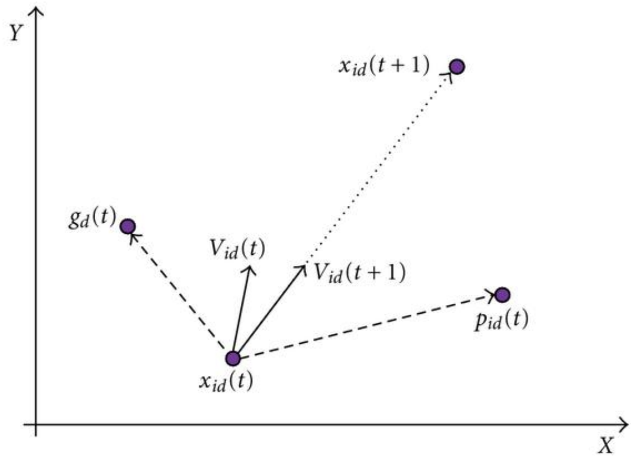


Figure 4. Moving a particle in PSO

*2.2 The proposed PSO*

To solve CVRP problem by PSO algorithm, the number of *n* customers, and the number of vehicles *m* are considered, which is taken as data from the input or text file. Now, for each particle, random permutations for numbers *1* to *n* are obtained, and *m+1* is zero in between, so that the capacity of customers in each category is less than the vehicle capacity. Of course, if it is not feasible, there is usually a problem in allocating customers to the last vehicle. Now, by implementing two algorithms, including the insert and exchange moves, the corresponding solution can be converted into a feasible answer. Therefore, based on the number of initial populations, considered *m1* here, the initial particle population can be created. Now, by obtaining the objective functions for this population, *m2* particles with the best objective function can be considered as *pbest* solutions, and the rest of the particles can be attributed to them. The method considered here is that the allocated particles to each group is not equal together, but based on particle power. In other words, the more *pbest* power, the more particles are attributed to it. Therefore, a number particles of each group is differ in size. The reason for using this idea in this part of the algorithm is that when quality of a particle is high, it is more likely that there are more high-quality answers in its neighborhood. Now, in order for particle movement to be performed based on the linear composition of *pbest* and *gbest* particles, a new strategy for implementing this concept in discrete problems such as CVRP is presented. The method of this work is that in two answers, customers find a subscription and are transferred to the desired particle in the same way. By modifying customers in the desired particle, the particle can be made possible. If there is no sharing between the two answers, then two *pbest* and *gbest* answers are firstly combined, and then the desired particle is combined with the result. Suppose particle = 0 1 2 3 4 0 5 6 0 7 8 9 0, pbest= 0 3 4 1 0 2 7 5 0 6 8 9 0, and gbest = 0 3 4 1 0 6 2 5 0 8 7 9 0. Customer set 0 1 4 3 0 is shared between these two *pbest* and *gbest* answers, now this set is added to the desired particle and the customer and additional zeroes are removed from the answer. If the answer is infeasible, insert and exchange algorithms are used.

After finishing the total movements of the particles, again the best of each category called *pbest* and the best *pbest* solutions called *gbest* will be updated. In other words, if the i*th* pbest bird is better than the gbest, it replaces the gbest and the bird number is stored as *i*. Otherwise, the gbest remains. Now, insert and exchange and 2-opt algorithms are used for all new *pbest* and *gbest* to improve the quality of the obtained answers. It should be noted that this method, due to the movement of particles and the use of local search algorithms, tries to pay full attention to the mechanisms of intensification and diversification, and thus can achieve excellent answers. This algorithm continues until the final condition of the algorithm is satisfied, which is the repetition of the main loop in a certain number. Pseudo-code of the proposed algorithm is presented in Fig. 5.

1- Input: *n, m*, i=1, characteristics of the customers and vehicles, including vehicle's capacity, customer's request, customer's coordinates, *m1* and *m2*.

2- produce *m1* solutions for the CVRP problem randomly, and find their objective functions.

1- Select *m2* numbers of the best solutions, and set as pbest solutions.
2- Attributed the other particles to *pbest* solutions based on the proposed mechanism, and create the *m2* group.

4- Select the best *pbest* solutions and called *gbest*.

5- while (i<=n)

- Move other particles to the pbest and gbest solutions according to the procedure.
- For each group, if the better solution than pbest is found, it will be updated.
- If a better solution than gbest is found, update it.
- Use local search algorithms for each updated pbest and gbest solutions.
- i=i+1

6- Output: the gbest solution with its objective function.

Figure 5. Steps of the proposed algorithm

**The local iterated algorithms:** A local iterated algorithm is one of the most famous algorithms for solving different optimization problems that can move in any iteration of the algorithm in the possible space of the problem, and get better solutions. So far, many techniques have been proposed by researchers for iterated methods because there is no direct method for finding the optimal solution for most optimization problems [36]. The main step in iterated methods is to make the next solution from the current one. If the stop criteria have been met that the algorithm ends and otherwise the algorithm will be repeated until these criteria are met. The method of searching for the sameness is a recurring procedure in which for each answer, a set of neighborhood is defined, and the best solution is selected from this collection. It should be noted that although these algorithms have been considered in the past decades, and have been used as main algorithms to solve optimization problems, nowadays with the advancement of meta-methods to solve these problems, the application of this kind of neighboring algorithms has been less. However, it should be noted that these algorithms are still used in most meta-heuristic algorithms as part of these algorithms or as a local search method for searching more space during the better response. For example, such algorithms can be used in the neighbor search area in the tabu search algorithm or after obtaining a new solution in each repetition of meta-heuristic algorithms, as a local improving algorithm to further improve the best solution. In this sub-section, three used neighbor search algorithms in the proposed algorithm such as insert, exchange and 2-opt algorithms are investigated.

In insert move method, a specific customer moves to another location on the same or different paths. It should be noted that in the following forms, it shows the warehouse and the customers, while the display is a special customer that has been moved on it. In exchange move, like Fig.s 7, two nodes are selected and change together. Finally, the 2-opt algorithm works by removing two edges from the cycle and re-connecting the two arcs in another way. This is done if the new tour is shortened. In addition, this operation is performed continuously so that no other dual recovery movement is found. Fig. 7 also shows this movement on a path.

## 3. PARAMETER SETTINGS

The solutions produced by the PPSO, like every meta-heuristic algorithm, were dependent on the seed used to generate the sequence of pseudo-random numbers and on the different values of the parameters of the algorithm. Many parameters exist in our proposed algorithm, and the values of these directly or indirectly affect the quality of the final solution. A parameter setting procedure is necessary to reach the best balance between the quality of the solutions obtained and the required computational attempt. We should mention that there is no way of defining the most effective values of the parameters.

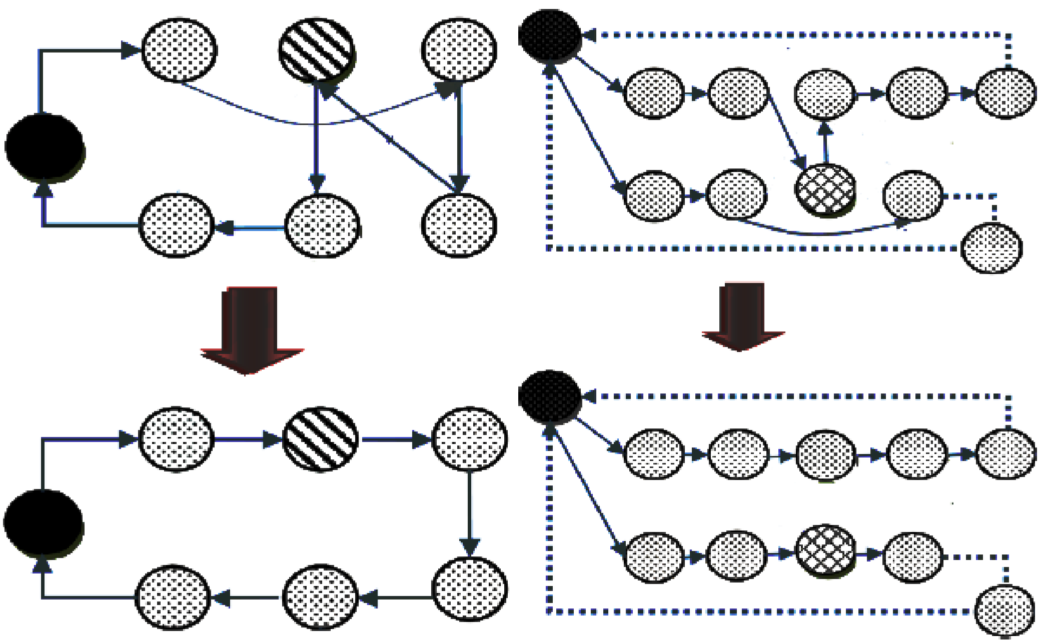

Figure 6. The insert move

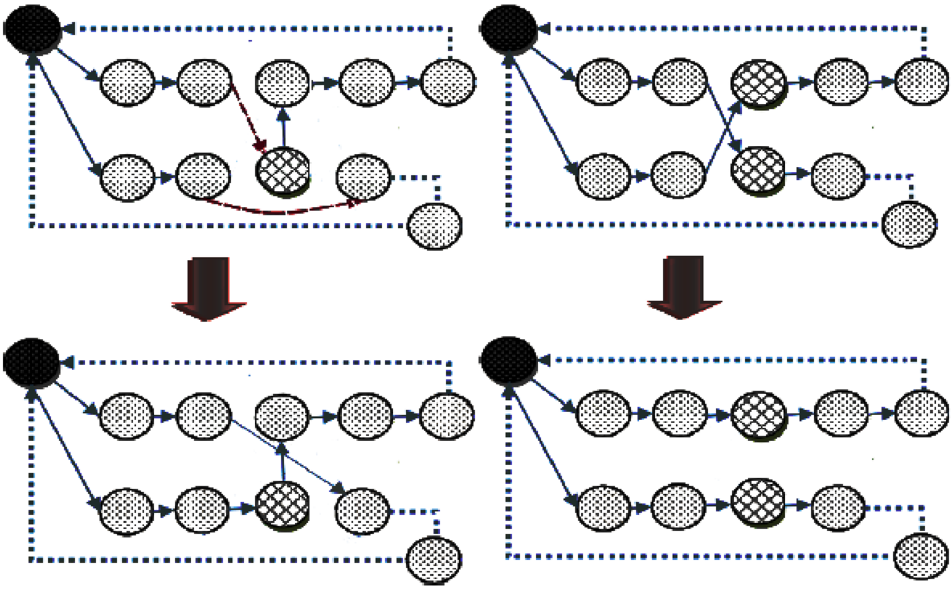

Figure 7. The change (left) and 2-opt (right) moves

Therefore in this paper, similar to many others, they are selected after thorough testing; we understand that the most influential (experimental) parameters in PPSO that directly affect the quality of the final solution are the number of populations, and the stop condition. Thus all of the parameter value shave been determined on the first instance i.e. C1 by the numerical experiments. Then to determine the value in the other instances: several alternative values for each parameter were tested while all the others were held constant, and by Taguchi method, the best values were selected giving the best computational results. After the selection of the final parameters, 10 different runs with the selected parameters were performed for each of the benchmark instances. The results confirm that our parameter setting worked well. It is also possible that better solutions could exist. The parameters of the PPSO and their values in different configurations are congregated in Table 1.

Table 1: Parameter setting for the proposed meta-heuristic method.

| **Parameter** | **Description** | **Problem number** | **Candidate value** | **Value** |
|---|---|---|---|---|
| m1 | The number of populations | C1,C2, C6, C7 | 10-15-…-50 | **45** |
| | | C3, C8,C11-14 | 10-15-…-50 | **55** |
| | | C4, C5, C9 ,C10 | 10-15-…-50 | **80** |
| m2 | The number of groups | C1–C14 | 5-10-15-20-25 | **10** |
| SC | The stop condition | C1–C14 | n/2,n,3n/2,2n | **n** |
| | | C2, C7 | n/2,n,3n/2,2n | **3n/2** |
| | | C3, C8,C12,C14 | n/2,n,3n/2,2n | **n** |
| | | C4, C5, C9 ,C10, C11, C13 | n/2,n,3n/2,2n | **n** |

## 4. COMPUTATIONAL RESULTS

Our proposed meta-heuristic method was programmed in Borland C, and executed on a Labtop core i3 with 4 Gb of RAM running Windows 7. The performance of the PPSO was tested on a set of benchmark problems for the CVRP issue. In this section, we first introduce the benchmark problems, and discuss the parameter setting of the proposed algorithm. Finally, the detailed computational results are discussed.

### *4.1 Benchmark problems*

In this section, the computational results of the proposed algorithm are presented in more detail, and the results are compared with other well-known meta-heuristic methods. It is important to note that the algorithm is written by Matlab7 language. For a more complete comparison, the proposed algorithm was implemented on two categories of standard VRP issues in the literature of the subject, presented in the following internet address.

http://www.vrp-rep.org/solutions.html

In the first category, 14 standard examples of Christofides are considered, firstly, they have a suitable combination of problems and range between 50 and 199 nodes, without storage of goods, and secondly, many algorithms have been tested on these examples. Therefore, a suitable comparison can be made between the PPSO algorithms and other proposed algorithms before and after 2010 in order to investigate the efficiency of the algorithm more accurately. It should also be added that among these 14 examples, the ten examples of C1-C10 include customers randomly distributed around the warehouse, while in the remaining four examples the customers are placed in the classes where the warehouse is not located in the center. Also, all of the intended examples have capacity limitations, but C6-C10, C13, C14 have capacity and route length limitations, and service time. In other words, in these examples, each vehicle cannot survey more than a certain amount of distance. The characteristics of these examples are shown in Table 2 along with the results of the proposed algorithm and the best known solutions (BKSs).

Table 2: Characteristics of the test problems and results of the proposed algorithm

| Instance | n | m | Q | D | PPSO | m-PPSO | CPU (sec) | BKS |
|---|---|---|---|---|---|---|---|---|
| C1 | 50 | 5 | 160 | ∞ | 524.61 | 5 | 94.485 | 524.61 |
| C2 | 75 | 10 | 140 | ∞ | 847.14 | 10 | 227.359 | 835.26 |
| C3 | 100 | 8 | 200 | ∞ | 712.36 | 8 | 249.625 | 826.14 |
| C4 | 150 | 12 | 200 | ∞ | 1066.89 | 12 | 470.1 | 1028.42 |
| C5 | 199 | 17 | 200 | ∞ | 1377.23 | 17 | 553.84 | 1291.29 |
| C6 | 50 | 6 | 160 | 200 | 524.61 | 5 | 65.628 | 555.43 |
| C7 | 75 | 11 | 140 | 160 | 860.50 | 11 | 176.14 | 909.68 |
| C8 | 100 | 9 | 200 | 230 | 860.67 | 8 | 255.719 | 865.94 |
| C9 | 150 | 14 | 200 | 200 | 1080.30 | 12 | 447.94 | 1162.55 |
| C10 | 199 | 18 | 200 | 200 | 1401.65 | 17 | 587.74 | 1395.85 |
| C11 | 120 | 7 | 200 | ∞ | 1174.12 | 7 | 386.49 | 1042.11 |
| C12 | 100 | 10 | 200 | ∞ | 840.637 | 10 | 223.579 | 819.56 |
| C13 | 120 | 11 | 200 | 720 | 1260.89 | 7 | 450.391 | 1541.14 |
| C14 | 100 | 11 | 200 | 1040 | 840.52 | 10 | 190.281 | 866.37 |

It should be noted that the PPSO algorithm is a highly competitive way to solve this problem that has been able to achieve high quality solutions. This algorithm has been able to achieve the best answers from the 14 examples in eight examples, and in only six examples, C2, C4, C5, C10, C11, and C12 have not been able to achieve BKS answers. Also, to show the efficiency of the algorithm, the percentage of deviation to the best answers obtained for each example is shown in Fig. 9. This percentage is obtained from the formula (3).

$$Gap = \frac{f(s) - f(s^*)}{f(s^*)} \times 100 \quad (3)$$

In this formula, $f(s)$ is the value obtained by the algorithm, while $f(s^*)$ represents the best value obtained for that problem so far. Therefore, it can be concluded that if the percentage of deviation for an arbitrary algorithm is obtained negative in the minimization problem, the obtained is better than the value of BKS, But if this positive value is obtained, it means that the algorithm has not been able to increase the quality of the answer.

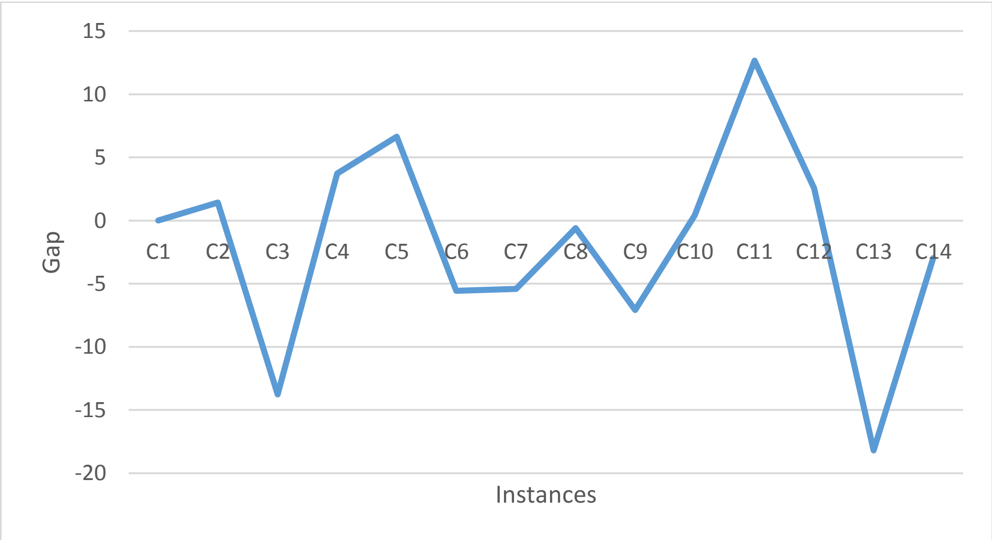


Figure 9. Percentage of deviation to the solutions obtained by proposed algorithm

*4.2 Results on benchmark instances*

The solutions obtained from comparing the proposed algorithm with some of the best methods available for VRP problem proposed before 2010 are shown in Table 3. In this table, each method is briefly described as follows.

SA: The simulated annealing [18]

TS: The tabu search [18]

GA: The genetic algorithm [37]

SS-ACO: The combined ACO method with local scatter method [38]

PSO: The particle swarm optimization [39]

GAPSO: The hybrid method of PSO and genetic algorithm [40]
PPSO: The proposed algorithm

The examples shown in this table can be divided into three categories: less than 100 (small), between 100 to 150 (average), and above 150 (large) customers. Therefore, the results of the algorithm on these three categories can be examined separately. In the first category, four examples of algorithms have been almost complete success and have been able to achieve the best answers ever found in 0.75 examples. This shows that algorithm has excellent efficiency in small examples. In the second categorym and in the six available instances, in 0.83 examples, the algorithm has been able to achieve the BKS answers. Therefore, it can be said that the algorithm has a very good efficiency for solving medium examples. Finally, in the third category, including four examples of algorithm, it has good performance and, like the first category, has been able to achieve the best answers in 0.75 examples.

Also, it can be noted that SA has only been able to produce optimal solutions in two examples, but the TS algorithm has been able to achieve BKS solutions in four examples. In this view, the GA algorithm has obtained worse solutions than the previous two algorithms and has only been able to reach the optimal solution in one example. Another algorithm that achieves similar results in this table is the ACO-SS algorithm, which has only been able to achieve BKS answers in two examples. On the other hand, the PSO algorithm, like the TS algorithm, has achieved the best answers in four examples, and SS_ACO produces better answers than the previous three algorithms. also, this algorithm has obtained weaker answers than the GAPSO algorithm because this algorithm has achieved BKS answers in ten examples. It should be noted that in this table, the PPSO algorithm has obtained the best answers, in eight examples as mentioned.

Table 3: Computational results for the benchmark of Christofides et al

| Instance | n | SA | TS | GA | SS_ACO | PSO | GAPSO | PPSO | PD | BKS |
|---|---|---|---|---|---|---|---|---|---|---|
| C1 | 50 | 528 | 524 | 524.61 | 524.61 | 524.61 | 524.61 | 524.61 | 0 | 524.61 |
| C2 | 75 | 838 | 844 | 849.77 | 835.26 | 844.42 | 835.26 | 847.41 | -1.45 | 835.26 |
| C3 | 100 | 829 | 835 | 840.72 | 830.14 | 829.40 | 826.14 | 712.36 | 13.77 | 826.14 |
| C4 | 150 | 1058 | 1052 | 1055.85 | 1038.20 | 1048.89 | 1028.42 | 1066.89 | -3.74 | 1028.42 |
| C5 | 199 | 1376 | 1354 | 1378.73 | 1307.18 | 1323.89 | 1294.21 | 1377.23 | -6.64 | 1291.45 |
| C6 | 50 | 555 | 555 | 560.29 | 559.12 | 555.43 | 555.43 | 524.61 | 5.55 | 555.43 |
| C7 | 75 | 909 | 913 | 914.13 | 912.68 | 917.68 | 909.68 | 860.50 | 5.41 | 909.68 |
| C8 | 100 | 866 | 866 | 872.82 | 869.34 | 867.01 | 865.94 | 860.67 | 0.61 | 865.94 |
| C9 | 150 | 1164 | 1188 | 1193.05 | 1179.4 | 1181.14 | 1163.41 | 1080.30 | 7.07 | 1162.55 |
| C10 | 199 | 1418 | 1422 | 1483.06 | 1410.26 | 1428.46 | 1397.51 | 1401.65 | -0.42 | 1395.85 |
| C11 | 120 | 1176 | 1042 | 1060.24 | 1044.12 | 1051.87 | 1042.11 | 1174.12 | -12.67 | 1042.11 |
| C12 | 100 | 826 | 819 | 877.8 | 824.31 | 819.56 | 819.56 | 840.64 | -2.57 | 819.56 |
| C13 | 120 | 1545 | 1547 | 1562.25 | 1556.52 | 1546.20 | 1544.57 | 1260.89 | 18.18 | 1541.14 |
| C14 | 100 | 890 | 866 | 872.34 | 870.26 | 866.37 | 866.37 | 840.52 | -2.98 | 866.37 |

Fig. 7 shows the average of the obtained answers for each algorithm. These answers show that the algorithm has much more efficiency than GA and SA algorithms, and has achieved better answers, while this algorithm has a very good competition compared to PSO, ACO-SS and TS algorithms. In other words, PPSO is located with a short distance from them and has been able to get better answers. Finally, the algorithm has been able to compete very closely with the combined GAPSO method and produce almost the same answers.

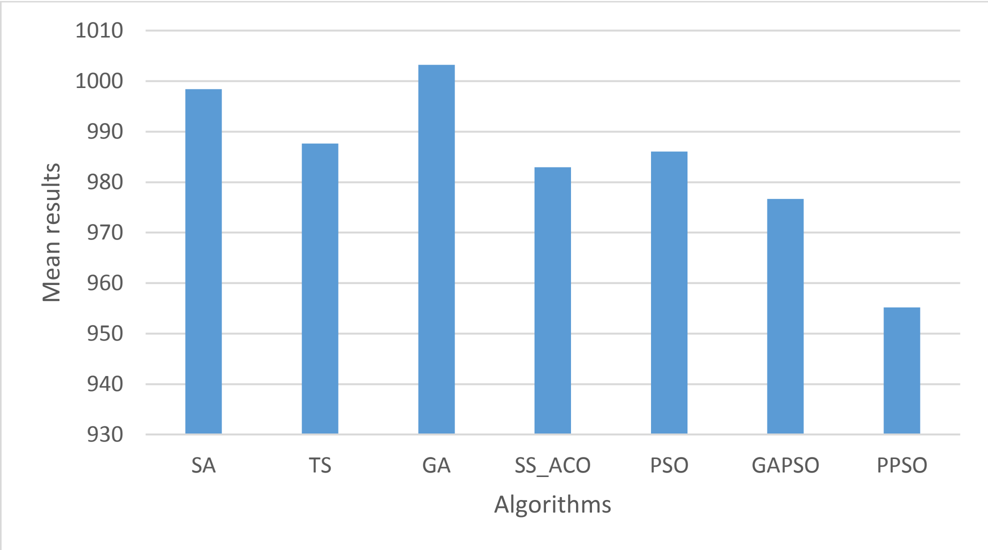


Figure 10. Comparing the mean results of the algorithms

Table 4 compares the results of the proposed algorithm on 14 examples, with those presented after 2010. These algorithms include the improved PSO (IPSO) [19], quantum-classic hybrid solution method (QCHSM) [41], gravitational emulation local search and genetic algorithm (GELSGA) [42], component-based heuristic algorithm (CBHA) [43], enhanced intelligent water drops and cuckoo search algorithm (EIWDCSA) [44], gravitational emulation local search algorithm (GELSA) [45], and variable neighborhood simulated annealing algorithm (VNSAA) [46]. It should be noted that among these seven algorithms, IPSO and QCHSM algorithms have been tested in only seven examples of C1-C5, C11 and C12, while VNSAA algorithm has been implemented in six examples: C1-C4, C11 and C12. Therefore, comparing the remaining five methods with these methods, although it can be investigated based on the average of the answers and Gaps, is not accurate enough. By comparing these three methods, it can be seen that the QCHSM algorithm is not of good quality so that the algorithm has not been able to achieve BKS answers in any of the seven examples. On the other hand, VNSAA and IPSO algorithms have close competition in six tested examples and only in two instances C3 and C4, IPSO algorithms have obtained better answers by minor difference, and the answers are the same in other examples. Compared to BKS results, it can be added that the IPSO algorithm has been able to achieve the best results in five from seven available examples. In addition, the proposed algorithm has been able to obtain comparative results compared to this algorithm. In other words, the IPSO algorithm has been able to achieve the best BKS answers in 5 examples, while the

PPSO algorithm has achieved BKS answers in just one example. On the other hand, unlike IPSO, the proposed algorithm has been able to improve BKS solutions in six examples.

Since the results of the remaining four algorithms of GELSGA, CBHA, EIWDCSA and GELSA, like the proposed algorithm, are presented in all 14 examples, these results can be compared in terms of obtaining BKSs, improving them and average of solutions. For this reason, in Fig. 11, the average results for 14 examples are presented by all algorithms. As can be concluded, although the algorithm is weaker than other algorithms in terms of the number of BKSs obtained, but if the criterion is considered to improve the best BKS solutions, then the proposed algorithm has achieved very high-quality results. Also, comparing the average of the answers in this form, it can be concluded that the two algorithms EIWDCSA and GELSA have obtained weaker answers compared to the other algorithms and similarly produced 976.79 for 14 examples. In addition, GELSGA and CBHA algorithms obtained 976.03 and 976.04 answers compared to the two mentioned algorithms. Finally, the best algorithm among these methods is PPSO, which has achieved the best result with a value of 955.15.

Table 4: Comparison of the results of the proposed algorithm on 14 examples, with those presented after 2010

| Instance | n | IPSO | QCHSM | VNSAA | GELSGA | CBHA | EIWDCSA | GELSA | PPSO | BKS |
|---|---|---|---|---|---|---|---|---|---|---|
| C1 | 50 | 524.61 | 556 | 524.61 | 524.61 | 524.61 | 524.61 | 524.61 | 524.61 | 524.61 |
| C2 | 75 | 835.26 | 926 | 835.26 | 835.26 | 835.26 | 835.26 | 835.26 | 847.14 | 835.26 |
| C3 | 100 | 826.14 | 905 | 826.24 | 826.14 | 826.14 | 826.14 | 826.14 | 712.36 | 826.14 |
| C4 | 150 | 1030.37 | 1148 | 1031.44 | 1028.42 | 1028.42 | 1028.42 | 1028.42 | 1066.89 | 1028.42 |
| C5 | 199 | 1297.63 | 1429 | - | 1291.29 | 1291.45 | 1294.21 | 1294.21 | 1377.23 | 1291.29 |
| C6 | 50 | - | - | - | 555.43 | 555.43 | 555.43 | 555.43 | 524.61 | 555.43 |
| C7 | 75 | - | - | - | 909.68 | 909.68 | 914.13 | 914.13 | 860.50 | 909.68 |
| C8 | 100 | - | - | - | 865.94 | 865.94 | 869.34 | 869.34 | 860.67 | 865.94 |
| C9 | 150 | - | - | - | 1162.55 | 1162.55 | 1162.55 | 1162.55 | 1080.30 | 1162.55 |
| C10 | 199 | - | - | - | 1395.85 | 1395.85 | 1395.85 | 1395.85 | 1401.65 | 1395.85 |
| C11 | 120 | 1042.11 | 1084 | 1042.11 | 1042.11 | 1042.11 | 1042.11 | 1042.11 | 1174.12 | 1042.11 |
| C12 | 100 | 819.56 | 828 | 819.56 | 819.56 | 819.56 | 819.56 | 819.56 | 840.637 | 819.56 |
| C13 | 120 | - | | - | 1541.14 | 1541.14 | 1541.14 | 1541.14 | 1260.89 | 1541.14 |
| C14 | 100 | - | | - | 866.37 | 866.37 | 866.37 | 866.37 | 840.52 | 866.37 |

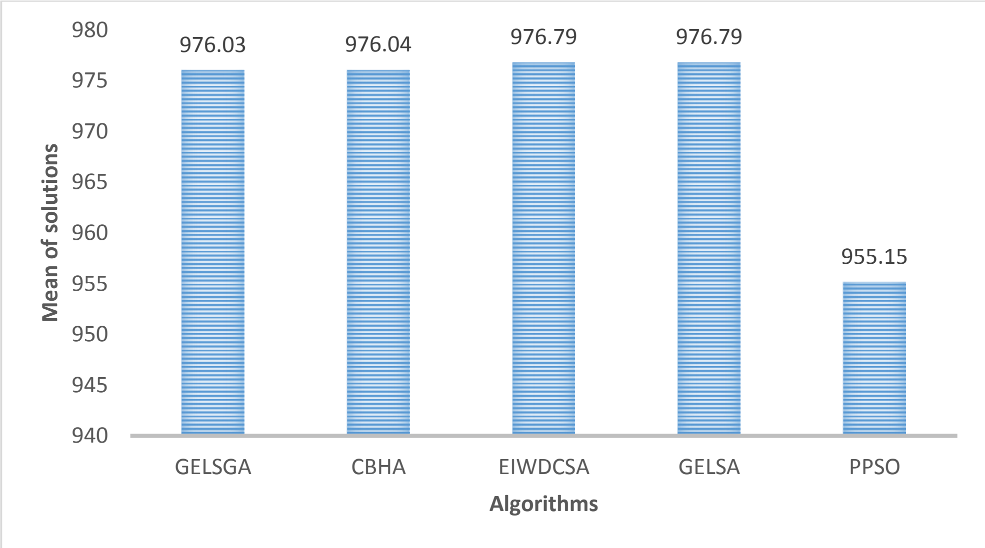

Figure 11. Comparing the mean results of the new meta-heuristic algorithms

## 5. CONCLUSIONS

In this paper, the modified hybrid PSO called PPSO was presented to solve CVRP efficiency in which several modifications were used to increase the efficiency of the algorithm. These modifications made the algorithm able to achieve better performance to escape from local optimal points and could achieve very good solutions. It seems that combining the relevant algorithm with other metaheuristic methods such as tabu search or using strong local algorithms such as Len-Kernighan method can have better results for the proposed algorithm. Besides, this algorithm can be used on other optimization problems such as vehicle routing problem with time windows and capacity clustering problem. Working on these ideas and applying them will be postponed to the next articles.

## APPENDIX A

*New Best Solutions*

The best solutions found by the proposed algorithm for the problems are presented in Tables A1-A14. In these Tables, Columns 2-5 show the route of each vehicle, the number of customers (NC), the sum of requests (SR) and the route cost (RC).

All the calculations have been performed with a precision of 64 bits and the total solution cost is presented with three or four decimal places.

Table A1

| | Route | NC | SR | RC |
|---|---|---|---|---|
| C1 | 1 19 14 42 41 20 43 18 5 48 1 | 9 | 157 | 109.056 |
| | 1 9 27 32 29 4 37 36 21 23 2 33 1 | 11 | 149 | 118.519 |
| | 1 7 15 26 25 44 8 24 49 28 1 | 9 | 152 | 98.4517 |
| | 1 12 3 30 22 17 51 35 31 10 39 1 | 10 | 159 | 99.3331 |
| | 1 13 38 45 16 46 34 40 11 50 6 47 1 | 11 | 160 | 99.2512 |
| Sum | | 50 | 777 | 524.611 |

Table A2

| | **Route** | **NC** | **SR** | **RC** |
|---|---|---|---|---|
| **C2** | 1 27 41 33 45 4 52 76 1 | 7 | 139 | 58.6108 |
| | 1 69 3 29 62 22 75 31 1 | 7 | 140 | 74.3784 |
| | 1 59 11 39 12 54 1 | 5 | 130 | 71.7338 |
| | 1 13 40 10 26 56 19 51 18 1 | 8 | 137 | 95.7482 |
| | 1 74 2 44 42 43 65 23 63 1 | 8 | 126 | 94.4481 |
| | 1 28 38 21 71 61 72 70 37 48 49 1 | 10 | 139 | 107.066 |
| | 1 17 50 25 57 24 64 34 7 1 | 8 | 140 | 92.69 |
| | 1 5 46 30 6 16 58 14 53 1 | 8 | 138 | 68.8575 |
| | 1 68 8 20 55 9 47 35 1 | 7 | 138 | 65.4722 |
| | 1 36 15 60 67 66 32 73 1 | 7 | 137 | 118.135 |
| **Sum** | | 75 | 1364 | 847.14 |

Table A3

| | **Route** | **NC** | **SR** | **RC** |
|---|---|---|---|---|
| **C3** | 1 51 80 79 35 36 72 66 67 21 33 1 | 10 | 153 | 98.1952 |
| | 1 28 70 2 71 32 11 63 89 8 49 9 1 | 11 | 158 | 75.8285 |
| | 1 90 61 84 85 6 62 86 92 17 45 15 1 | 11 | 174 | 64.3106 |
| | 1 29 77 78 30 25 55 56 26 5 40 57 24 68 1 | 13 | 191 | 105.803 |
| | 1 53 19 83 20 65 50 37 48 47 46 18 87 39 1 | 13 | 195 | 140.328 |
| | 1 27 13 81 69 4 34 82 10 52 31 91 64 12 1 | 13 | 200 | 88.2475 |
| | 1 54 59 41 22 74 73 75 76 23 42 16 44 43 88 58 3 1 | 16 | 195 | 90.9776 |
| | 1 7 97 100 60 94 99 101 38 93 98 96 95 14 1 | 13 | 192 | 48.6732 |
| **Sum** | | 100 | 1458 | 712.3636 |

Table A4

| | **Route** | **NC** | **SR** | **RC** |
|---|---|---|---|---|
| **C4** | 1 103 7 58 24 70 87 44 100 115 62 8 113 49 139 47 1 | 15 | 198 | 118.613 |
| | 1 39 63 51 131 35 75 80 22 119 17 127 79 1 | 12 | 176 | 71.372 |
| | 1 12 101 3 84 132 21 60 4 102 52 120 1 | 11 | 185 | 75.3747 |
| | 1 91 125 126 107 74 118 90 40 76 106 31 105 10 1 | 13 | 196 | 108.243 |

| | | | | |
|---|---|---|---|---|
| | 1 111 135 68 14 137 41 89 65 20 95 42 67 142 88 143 1 | 15 | 197 | 110.527 |
| | 1 104 109 53 16 124 72 123 55 11 50 77 6 1 | 12 | 174 | 72.0972 |
| | 1 19 56 112 136 144 146 64 1 | 7 | 186 | 59.0044 |
| | 1 38 138 45 108 46 73 34 92 66 94 43 93 151 149 148 18 1 | 16 | 189 | 110.719 |
| | 1 140 48 57 147 5 150 110 145 13 1 | 9 | 156 | 40.7791 |
| | 1 134 15 59 26 96 97 25 98 99 133 69 1 | 11 | 187 | 93.7306 |
| | 1 128 54 130 30 129 85 36 86 37 116 122 117 29 71 23 1 | 15 | 195 | 122.648 |
| | 1 78 33 2 121 81 32 83 141 114 27 9 61 82 28 1 | 14 | 196 | 83.783 |
| **Sum** | | 150 | 2235 | 1066.89 |

Table A5

| | **Route** | **NC** | **SR** | **RC** |
|---|---|---|---|---|
| **C5** | 1 31 60 6 104 139 167 21 125 155 16 80 154 1 | 12 | 198 | 95.1296 |
| | 1 188 98 10 162 111 76 163 32 81 132 190 170 51 169 127 1 | 15 | 199 | 102.999 |
| | 1 96 152 56 45 107 73 119 136 93 149 164 26 57 33 1 | 14 | 200 | 115.463 |
| | 1 88 5 112 28 84 14 124 129 71 20 177 47 1 | 12 | 195 | 76.4849 |
| | 1 113 195 159 199 54 196 194 34 62 1 | 9 | 183 | 41.5824 |
| | 1 153 49 172 48 156 37 123 38 89 30 55 1 | 11 | 187 | 84.1162 |
| | 1 8 133 181 70 85 135 86 165 171 12 109 1 | 11 | 199 | 104.406 |
| | 1 192 2 185 191 44 200 137 197 187 94 29 102 1 | 12 | 195 | 70.6578 |
| | 1 198 193 161 116 91 42 144 69 43 114 92 142 23 67 1 | 14 | 197 | 87.8428 |
| | 1 4 182 148 19 147 74 146 25 108 64 50 183 118 1 | 13 | 199 | 87.8451 |
| | 1 128 35 66 168 180 100 59 46 99 126 1 | 10 | 169 | 51.3963 |
| | 1 87 157 115 143 138 90 186 63 117 145 75 184 24 105 106 1 | 15 | 200 | 126.902 |
| | 1 61 150 52 36 15 134 178 79 179 103 9 176 1 | 12 | 191 | 76.4123 |
| | 1 13 58 40 110 131 41 77 18 1 | 8 | 166 | 56.4586 |
| | 1 140 121 173 22 174 175 83 122 141 95 65 3 158 1 | 13 | 193 | 69.4304 |
| | 1 7 97 68 160 17 189 1 | 6 | 122 | 38.1157 |
| | 1 82 72 130 11 78 120 39 166 53 151 101 27 1 | 12 | 193 | 91.9855 |
| **Sum** | | 199 | 3186 | 1377.23 |

Table A6

| | **Route** | **NC** | **SR** | **RC** |
|---|---|---|---|---|
| **C6** | 1 19 14 42 41 20 43 18 5 48 1 | 9 | 157 | 109.056 |
| | 1 9 27 32 29 4 37 36 21 23 2 33 1 | 11 | 149 | 118.519 |
| | 1 7 15 26 25 44 8 24 49 28 1 | 9 | 152 | 98.4517 |
| | 1 12 3 30 22 17 51 35 31 10 39 1 | 10 | 159 | 99.3331 |
| | 1 13 38 45 16 46 34 40 11 50 6 47 1 | 11 | 160 | 99.2512 |
| **Sum** | | 50 | 777 | 524.611 |

Table A7

| | Route | NC | SR | RC |
|---|---|---|---|---|
| C7 | 1 48 37 70 72 61 71 21 38 6 30 1 | 10 | 136 | 103.468 |
| | 1 47 68 1 | 2 | 57 | 22.5655 |
| | 1 41 33 45 4 18 1 | 5 | 109 | 50.889 |
| | 1 31 49 22 62 29 75 1 | 6 | 124 | 77.4163 |
| | 1 73 40 26 56 51 19 25 50 17 52 1 | 10 | 136 | 118.336 |
| | 1 54 12 67 66 39 1 | 5 | 129 | 77.1576 |
| | 1 27 59 11 32 10 13 1 | 6 | 135 | 81.4167 |
| | 1 76 69 3 63 2 34 7 1 | 7 | 138 | 57.3679 |
| | 1 9 55 20 60 15 36 8 1 | 7 | 127 | 94.6231 |
| | 1 64 24 57 42 44 43 65 23 74 1 | 9 | 133 | 111.245 |
| | 1 35 53 28 14 58 16 46 5 1 | 8 | 140 | 66.0115 |
| **Sum** | | 75 | 1364 | 860.496 |

Table A8

| | Route | NC | SR | RC |
|---|---|---|---|---|
| C8 | 1 19 9 47 48 37 50 65 64 91 33 11 32 28 1 | 13 | 188 | 133.942 |
| | 1 14 88 43 15 44 16 58 3 42 23 76 75 74 41 54 1 | 15 | 184 | 114.687 |
| | 1 59 22 73 57 24 68 40 26 56 5 27 1 | 11 | 189 | 107.641 |
| | 1 89 63 12 20 49 83 8 53 1 | 8 | 123 | 81.3688 |
| | 1 95 60 100 94 86 92 101 38 99 93 98 96 1 | 12 | 197 | 60.5233 |
| | 1 90 61 84 46 18 87 39 45 17 62 85 6 97 7 1 | 14 | 195 | 119.107 |
| | 1 70 71 31 21 67 66 72 36 10 82 4 78 13 1 | 13 | 197 | 123.593 |
| | 1 2 51 52 34 80 79 35 30 25 55 81 69 77 29 1 | 14 | 185 | 119.807 |
| **Sum** | | 100 | 1458 | 860.669 |

Table A9

| | Route | NC | SR | RC |
|---|---|---|---|---|
| C9 | 1 109 64 146 110 144 5 150 147 145 13 1 | 10 | 196 | 48.3727 |
| | 1 127 22 80 129 85 36 86 37 116 122 117 4 60 21 1 | 14 | 198 | 137.808 |
| | 1 18 148 143 88 149 136 142 151 20 65 43 94 66 93 45 138 1 | 16 | 198 | 103.297 |
| | 1 28 82 139 49 61 9 81 121 2 120 78 1 | 11 | 162 | 71.0206 |
| | 1 47 103 7 58 113 62 27 114 141 83 32 29 71 102 23 52 33 1 | 17 | 182 | 106.336 |
| | 1 6 11 55 106 76 40 90 118 74 107 126 124 91 1 | 13 | 194 | 109.766 |
| | 1 48 19 111 134 26 96 59 15 69 140 1 | 10 | 181 | 65.5119 |
| | 1 104 72 123 125 34 73 92 46 108 16 53 38 1 | 12 | 147 | 71.7171 |
| | 1 12 101 3 84 132 30 130 54 128 17 79 1 | 11 | 186 | 62.0942 |
| | 1 133 99 25 97 98 87 44 100 115 8 70 24 1 | 12 | 197 | 129.961 |
| | 1 57 112 67 42 95 89 41 137 14 68 135 56 1 | 12 | 196 | 102.883 |
| | 1 77 50 31 105 35 75 119 51 131 10 63 39 1 | 12 | 198 | 72.535 |
| **Sum** | | 150 | 2235 | 1080.3 |

Table A10

| | Route | NC | SR | RC |
|---|---|---|---|---|
| **C10** | 1 96 152 107 19 147 136 93 149 164 57 10 131 51 1 | 13 | 196 | 119.605 |
| | 1 118 64 108 25 146 74 148 182 45 56 4 1 | 11 | 175 | 80.2484 |
| | 1 153 140 173 175 156 37 139 38 89 154 28 180 59 112 1 | 14 | 198 | 90.7934 |
| | 1 168 20 178 134 15 85 135 86 165 171 166 120 1 | 12 | 199 | 125.518 |
| | 1 176 177 9 103 133 181 36 179 79 71 129 124 14 84 100 1 | 15 | 200 | 89.5574 |
| | 1 113 187 23 142 92 43 144 91 42 191 44 198 197 67 1 | 14 | 194 | 77.5223 |
| | 1 189 17 183 50 75 145 117 63 24 68 160 97 1 | 12 | 194 | 76.6734 |
| | 1 99 30 80 16 155 125 21 167 123 48 172 121 1 | 12 | 197 | 99.3201 |
| | 1 7 106 196 54 199 194 34 1 | 7 | 105 | 36.5585 |
| | 1 61 27 150 52 8 47 66 35 128 1 | 9 | 172 | 52.1679 |
| | 1 105 184 161 116 186 90 138 69 114 143 115 157 94 87 1 | 14 | 199 | 119.985 |
| | 1 13 110 40 58 190 76 163 32 132 81 11 170 169 82 1 | 14 | 199 | 93.2782 |
| | 1 3 158 102 29 65 95 141 122 83 174 22 1 | 11 | 182 | 65.074 |
| | 1 18 77 41 188 98 162 111 26 119 73 33 1 | 11 | 200 | 75.2899 |
| | 1 5 88 126 46 104 6 60 49 31 55 1 | 10 | 191 | 57.4456 |
| | 1 195 192 2 137 200 185 193 159 62 1 | 9 | 185 | 52.2353 |
| | 1 127 72 130 78 39 53 12 109 70 151 101 1 | 11 | 200 | 90.3782 |
| **Sum** | | 199 | 3186 | 1401.65 |

Table A11

| | Route | NC | SR | RC |
|---|---|---|---|---|
| **C11** | 1 105 104 69 80 81 57 61 64 67 65 63 62 66 58 55 53 99 100 1 | 18 | 200 | 216.011 |
| | 1 101 111 38 39 42 45 30 33 36 37 35 32 31 34 28 25 23 13 9 109 1 | 20 | 200 | 229.513 |
| | 1 103 102 98 116 22 21 24 27 29 26 20 17 18 110 95 94 97 1 | 17 | 197 | 178.488 |
| | 1 121 68 70 71 72 75 73 76 79 78 77 74 108 107 106 0 | 15 | 194 | 135.815 |
| | 1 40 43 48 47 50 51 52 49 46 44 41 60 59 56 54 117 1 | 16 | 199 | 234.679 |
| | 1 120 82 3 2 4 5 6 11 12 16 15 14 10 8 7 83 1 | 16 | 198 | 126.614 |
| | 1 96 88 93 90 92 91 115 19 119 84 114 118 85 113 86 87 112 89 1 | 18 | 187 | 52.9974 |
| **Sum** | | 120 | 1375 | 1174.12 |

Table A12

| | Route | NC | SR | RC |
|---|---|---|---|---|
| | 1 44 43 42 41 45 46 47 49 51 52 53 50 48 1 | 13 | 160 | 66.4736 |
| | 1 100 101 98 94 93 95 96 97 99 1 | 9 | 190 | 95.9431 |
| | 1 60 61 59 57 54 55 56 58 1 | 8 | 200 | 101.883 |
| | 1 82 79 77 72 71 74 78 80 81 73 62 65 69 70 1 | 14 | 200 | 137.019 |
| **C12** | 1 68 66 64 75 63 67 1 | 6 | 150 | 43.5905 |
| | 1 14 18 19 20 16 17 15 13 11 1 | 9 | 200 | 96.0398 |
| | 1 76 2 3 5 7 10 12 9 8 4 6 1 | 11 | 170 | 56.1748 |
| | 1 24 27 29 31 35 37 40 39 38 36 32 1 | 11 | 180 | 95.9925 |
| | 1 92 90 89 86 85 83 84 87 88 91 1 | 10 | 170 | 76.0696 |
| | 1 22 23 26 28 30 34 33 25 21 1 | 9 | 190 | 71.4518 |
| **Sum** | | 100 | 1810 | 840.637 |

Table A13

| | Route | NC | SR | RC |
|---|---|---|---|---|
| | 1 110 38 42 45 47 48 50 51 52 49 43 40 39 116 98 94 1 | 16 | 197 | 191.777 |
| | 1 108 74 77 78 79 73 76 75 72 71 70 68 121 120 1 | 14 | 200 | 145.652 |
| | 1 90 91 19 119 109 27 29 33 36 30 37 35 32 31 34 28 25 23 14 10 8 7 87 1 | 23 | 200 | 210.556 |
| **C13** | 1 97 117 111 54 56 59 63 65 67 64 61 57 81 80 69 104 105 107 106 1 | 19 | 193 | 215.699 |
| | 1 96 95 41 44 46 60 66 62 58 55 53 99 101 100 102 103 1 | 16 | 197 | 209.533 |
| | 1 112 113 118 114 84 6 11 12 15 16 5 4 2 3 82 83 89 1 | 17 | 192 | 116.185 |
| | 1 86 85 9 13 18 17 20 26 24 21 22 115 92 93 88 1 | 15 | 196 | 171.516 |
| **Sum** | | 120 | 1375 | 1260.89 |

Table A14

| | Route | NC | SR | RC |
|---|---|---|---|---|
| | 1 24 27 29 31 35 37 40 39 38 36 32 1 | 11 | 180 | 95.9925 |
| | 1 92 90 89 86 85 83 84 87 88 91 1 | 10 | 170 | 76.0696 |
| | 1 22 23 26 28 30 34 33 25 21 1 | 9 | 190 | 71.4518 |
| | 1 11 9 10 7 5 3 2 76 6 4 8 1 | 11 | 170 | 574463 |
| **C14** | 1 44 43 42 41 45 46 47 49 52 51 53 50 48 1 | 13 | 160 | 64.8075 |
| | 1 60 61 59 57 54 55 56 58 1 | 8 | 200 | 101.883 |
| | 1 99 97 96 95 93 94 98 101 100 1 | 9 | 190 | 95.9431 |
| | 1 14 18 19 20 16 17 15 13 12 1 | 9 | 200 | 96.3207 |
| | 1 70 69 65 62 73 81 80 78 74 71 72 77 79 82 1 | 14 | 200 | 137.019 |
| | 1 68 66 64 75 63 67 1 | 6 | 150 | 43.5905 |
| **Sum** | | 100 | 1810 | 840.523 |

## REFERENCES


1. Ashouri M, Yousefikhoshbakht M. A Combination of meta-heuristic and heuristic algorithms for the VRP, OVRP and VRP with Simultaneous Pickup and Delivery, BRAIN, *Broad Res Artific Intell Neurosci* 2017; **8**(2): 81-95.
2. Yousefikhoshbakht M, Dolatnejad A, Didehvar F, Rahmati F. A modified column generation to solve the heterogeneous fixed Fleet open vehicle routing problem, *J Eng* 2016.
3. Mahmoodi Darani N, Ahmadi V, Saadati Eskandari Z, Yousefikhoshbakht M. Solving the capacitated clustering problem by a combined meta-heuristic algorithm, *J Adv Comput Res* 2013; **4**(1): 89-100.
4. Maleki F, Yousefikhoshbakht M. A hybrid algorithm for the open vehicle routing problem, *Int J Optim Civil Eng* 2019; **9**(2): 355-71.
5. Soleimani H, Chaharlang Y, Ghaderi H. Collection and distribution of returned-remanufactured products in a vehicle routing problem with pickup and delivery considering sustainable and green criteria, *J Clean Product* 2018; **172**: 960-70.
6. Zhao PX, Luo WH, Han X. Time-dependent and bi-objective vehicle routing problem with time windows, *Adv Product Eng Manag* 2019; **14**(2): 201-12.
7. Yousefikhoshbakht M, Didehvar F, Rahmati F. A mixed integer programming formulation for the heterogeneous fixed fleet open vehicle routing problem, *J Optim Indust Eng* 2015; **8**(18): 37-46.
8. Yousefikhoshbakht M, Didehvar F, Rahmati F. An effective rank based ant system algorithm for solving the balanced vehicle routing problem, *Int J Indust Eng* 2016; **23**(1): 13-25.
9. Dantzig G, Ramser J. The truck dispatching problem, *Manag Sci* 1959, **6**: 80-91.
10. Fisher M, Jaikumar R. A generalized assignment heuristic for vehicle routing, *Networks* 1981; **11**: 109-24.
11. Christofides N, Mingozzi A, Toth P. The vehicle routing problem combinatorial optimization, Wiley, Chichester, (N. Christofides, A. Mingozzi, P. Toth and C. Sandi, Eds), 1979; 315-38.
12. Beheshti AK, Hejazi SR. A novel hybrid column generation-metaheuristic approach for the vehicle routing problem with general soft time window, *Inform Sci* 2015; **316**: 598-615.
13. Dayarian I, Crainic TG, Gendreau M, Rei W. A column generation approach for a multi-attribute vehicle routing problem, *European J Operat Res* 2015; **241**(3): 888-906.
14. Kheirkhah A, Navidi H, Bidgoli MM. An improved benders decomposition algorithm for an arc interdiction vehicle routing problem, *IEEE Transact Eng Manag* 2015; **63**(2): 259-73.
15. Kaveh A, Dadras A. A novel meta-heuristic optimization algorithm: thermal exchange optimization, *Adv Eng Softw* 2017; **110**: 69-84.
16. Kaveh A, Farhoudi N. A new optimization method: Dolphin echolocation, *Adv Eng Softw* 2013; **59**: 53-70.
17. Kaveh A, Mahdavi VR. Colliding bodies optimization: a novel meta-heuristic method, *Comput Struct* 2014; 139: 18-27.

18. Osman LH. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Ann Operat Res* 1993; **41**: 421-51.
19. Alinezhad H, Yaghoubi ÙS, Hoseini Motlagh SM, Allahyari S, Saghafi Nia M. An improved particle swarm optimization for a class of capacitated vehicle routing problems, *Int J Transport Eng* 2018; **5**(4): 331-47.
20. Soman JT, Patil RJ. A scatter search method for heterogeneous fleet vehicle routing problem with release dates under lateness dependent tardiness costs, *Expert Syst Applicat* 2020; **150**: 113-302.
21. Brandão J. A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem, *European J Operat Res* 2020; **284**(2): 559-571.
22. Yousefikhoshbakht M, Didehvar F, Rahmati F. An efficient solution for the vrp by using a hybrid elite ant system. Int. J. *Comput. Commun.* Control 2014; **9**(3):340-347.
23. Nakhaei F, Irannajad M, Yousefikhoshbakht M. Simultaneous optimization of flotation column performance using genetic evolutionary algorithm. *Physicochem. Probl. Miner. Process.* 2016; 52(2): 874–893.
24. Nakhaei F, Irannajad M, Yousefikhoshbakht M. Flotation column performance optimisation based on imperialist competitive algorithm. *Int. J. Min. Geo-Eng* 2016; **7**(1): 1-7.
25. Sedighpour M, Ahmadi V, Yousefikhoshbakht M, Didehvar F, Rahmati F. Solving the open vehicle routing problem by a hybrid ant colony optimization. *Kuwait J Sci* 2014; **29**: 41(3).
26. Zafari A, Tashakori S, Yousefi Khoshbakht M. A Hybrid Effective Genetic Algorithm for Solving the Vehicle Routing Problem. *Int J Ind Eng Prod Res* 2010; **21**(2): 63-76.
27. Yousefikhoshbakht M, Mahmoodabadi E, Sedighpour M. A modified elite ACO based avoiding premature convergence for traveling salesmen problem. *J IND ENG INT* 2011: **7**(15) 68-75.
28. Yousefikhoshbakht M, Malekzadeh N, Sedighpour M. Solving the traveling salesman problem based on the genetic reactive bone route algorithm whit ant colony system. *Int J Prod Manage Eng* 2016; **4**(2): 65-73.
29. Eskandaria ZS, Yousefikhoshbakhtb M. Solving the vehicle routing problem by an effective reactive bone route algorithm. *Trans Res* 2012; **2**(1): 39.
30. Yousefi Khoshbakht, M. and Zafari, A., 2008, October. A new ant colony algorithm for solving multiple traveling salesman problem. *In The 2th Joint Congress on Intelligent and Fuzzy Systems*, pp. 28-30.
31. Kennedy J, Eberhart R. Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks* 1995, Perth, Australia, pp. 1942-1948.
32. Zhang X, Liu H, Tu L. A modified particle swarm optimization for multimodal multi-objective optimization, *Eng Applicat Artificial Intell* 2020; **95**: 103905.
33. Karim AA, Isa NAM, Lim WH. Modified particle swarm optimization with effective guides, *IEEE Access* 2020: **8**: 188699-725.
34. Liu H, Zhang XW, Tu LP. A modified particle swarm optimization using adaptive strategy, *Expert Syst Applicat* 2020; 113-353.

35. Mirjalili S, Dong JS, Lewis A, Sadiq AS. *Particle Swarm Optimization: Theory, Literature Review, and Application in Airfoil Design*, In Nature-Inspired Optimizers 2020, Springer, Cham, pp. 167-184.
36. Fan SKS, Jen CH. An enhanced partial search to particle swarm optimization for unconstrained optimization, Mathematics 2019; **7**: 357.
37. Berger J, Barkaoui M. A hybrid genetic algorithm for the capacitated vehicle routing problem, in: P*roceedings of the International Genetic and Evolutionary Computation Conference – GECCO03, LNCS 2723*, 2020; pp. 646–656.
38. Zhang X, Tang L. A new hybrid ant colony optimization algorithm for the vehicle routing problem, *Pattern Recog Letter* 2009; **30**(9): 848-55.
39. Ai TJ, Kachitvichyanukul V. A particle swarm optimization for the heterogeneous fleet vehicle routing problem, *Int J Logistic SCM Syst* 2009, **3**(1): 32-9.
40. Marinakis Y, Marinaki M. A hybrid genetic – particle swarm optimization algorithm for the vehicle routing problem, *Expert Syst Applicat* 2010; **37**(2): 1446-55.
41. Feld S, Roch C, Gabor T, Seidel C, Neukart F, Galter I, Linnhoff-Popien C. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer, *Front ICT* 2019; **6**: 13.
42. Hosseinabadi AAR, Slowik A, Sadeghilalimi M, Farokhzad M, Sangaiah AK. An ameliorative hybrid algorithm for solving the capacitated vehicle routing problem, *IEEE Access* 2019; **7**: 175454-65.
43. Vidal T, Crainic TG, Gendreau M, Prins C. A unified solution framework for multi-attribute vehicle routing problems, *European J Operat Res* 2014; **234**(3): 658-73.
44. Teymourian E, Kayvanfar V, Komaki GM, Zandieh M. Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem, *Information Sciences* 2016; **334**: 354-78.
45. Hosseinabadi AAR, Rostami NSH, Kardgar M, Mirkamali S, Abraham A. A new efficient approach for solving the capacitated vehicle routing problem using the gravitational emulation local search algorithm, *Appl Mathemat Modell* 2017: **49**, 663-79.
46. Xiao Y, Zhao Q, Kaku I, Mladenovic N (2014). Variable neighbourhood simulated annealing algorithm for capacitated vehicle routing problems, *Eng Optim* 2017; **46**(4): 562-79.