A HYBRID IMPERIALIST COMPETITIVE ALGORITHM FOR THE BALANCED VEHICLE ROUTING PROBLEM

M. Sedighpour[1] and M. Yousefikhoshbakht[2, *, †]
¹Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran
²Department of Mathematics, Faculty of Sciences, Bu-Ali Sina University, Hamedan, Iran

ABSTRACT

The balanced vehicle routing problem (BVRP) is one of the most famous research problems in operations, which has a very important position in combination optimization problems. In this problem, a fleet of vehicles with capacity Q starts moving from a node called the warehouse and returns to it after serving customers, provided that they visit each customer only once and never exceed the capacity Q. The goal is to minimize the paths traveled by vehicles provided that the distances traveled by the vehicles are the same as possible, for more justice in working time and income. This article presents the application of a hybrid imperialist competitive algorithm (HICA) to solve the problem. Unlike other optimization methods, this method is inspired by the socio-political process of societies and uses the competition between colonizing and colonized countries to reach the solution. To test the effectiveness of the algorithm, a set of standard examples are considered and the algorithm is implemented on it. The calculation results on these examples, which have a size of 50 to 200, show that the proposed algorithm has been able to compete well with well-known meta-heuristic algorithms in terms of the quality of the answers. In addition, the solutions close to the best answers obtained so far are generated for most of the examples.

Keywords: Balanced vehicle routing problem; meta-heuristic; rank based ant system; NP-hard, Metaheuristics.

1. INTRODUCTION

The Vehicle Routing Problem (VRP) is one of the most important research issues in

---

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

transportation planning and operations, which is receiving a lot of attention from researchers and scientists today. In this problem, several vehicles start moving from a certain node called the goods warehouse at the same time and return to the same node after meeting the customers, provided that each node is visited by only one of these vehicles and each vehicle does not load from its capacity along the way. The aim is to determine the minimum displacement cost for the used vehicles according to the limitations of the problem. For example, Figure 1 shows two examples of solving a vehicle routing problem with a number of nodes and vehicles.
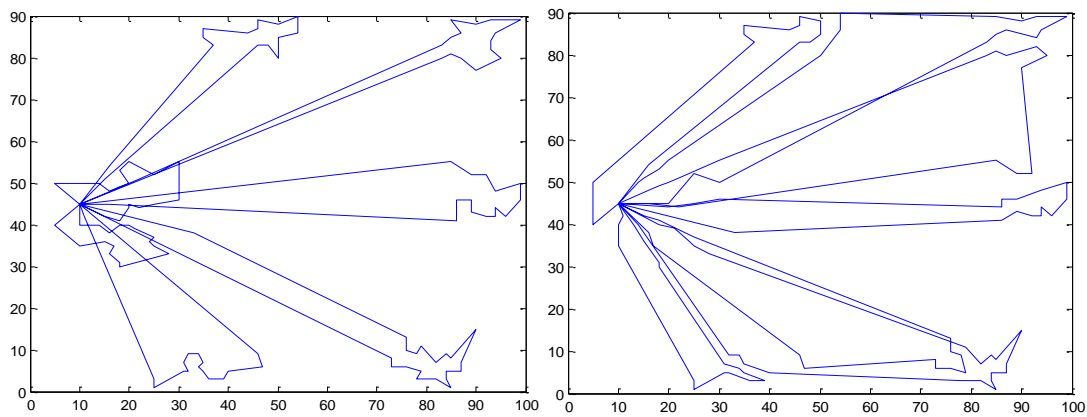

Figure 1. Two feasible solutions of VRPs

One of the factors that has caused this problem to be called an important problems of combined optimization is the application of this problem and its many extensions such as the classical mode [1,2], open [3], simultaneous pickup and delivery [4], heterogeneous fixed fleet [5], open heterogeneous fixed fleet [6], and so on in the real world. For example, suppose that a factory is able to reduce the length of the route to deliver goods to customers or the number of vehicles and, as a result, the cost of transportation. As a result, the relevant company will increase its competitive power in front of other similar companies and will expand the market of its goods and will eventually earn more profit. Since VRP problem was studied in the 60s, many modes of it were derived based on the different applications it had in the real world. The diversity of such problems is so great that their classification and the expression of the different states in which they occur are very difficult and time-consuming [7]. Therefore, only its capacitive state is considered here. This problem was first introduced by Dantzig et al. At that stage, they paid attention to the formulation of the problem and proposed a mathematical algorithm for it in the form of the problem of routing diesel trucks from the refueling terminal to the diesel demand stations [8].

In addition, they showed that use of management methods and optimization topics in transportation has a significant effect in reducing the final cost of goods. In this article, a method to find a solution close to the optimum was presented by these same people. In addition, in this problem, it was assumed that the capacity of the vehicles is equal to infinity and the duration of stopping at the nodes is considered to be zero. Therefore, selecting the set of customers assigned to each vehicle and determining the return route and the priority and delay of referring to them are two questions that must be answered in this issue. In other

words, in addition to categorizing customers and allocating each customer to a specific vehicle, car routing should also be done. Therefore, this primary problem is the problem of multiple traveling salesman problem, where there is no limit to the capacity of vehicles.

In general, there are many methods to solve this problem, which can be divided into two big categories of exact and approximate algorithms. In exact algorithms, the optimal solution of the problem can finally be achieved, but the algorithm must spend more time and, as a result, more cost to reach this solution. For example, among these types of algorithms, we can mention learning-based branch-and-price algorithm [9], minimum-k tree method [10] and branch and cut [11]. On the other hand, because the is NP-complete [12], therefore, with the increase in the size of the problem, the number of possible solutions starts to grow suddenly and with a very high acceleration, and the algorithm cannot achieve the optimal solution of the problem in an acceptable time. Therefore, for such problems, approximate algorithms, classified into two categories of heuristic and meta-heuristic algorithms, were given more attention and many versions of these algorithms were proposed (Figure 2).

Heuristic algorithms, unlike exact methods, arrive at the solution in a short time, although the corresponding solution may be less accurate than the solution of exact algorithms according to the size of the problem. It is important to note that innovative algorithms do not have an effective way to escape from local optimal points and most of the time they get stuck in local optima. For example, we can refer to the saving algorithm and its modification types [13], as well as Benders algorithm [14]. In the meta-heuristic methods that have been considered for almost 30 years, the answers are obtained more in the same time than the heuristic algorithms and less than the exact methods because the execution rate of the algorithm, unlike the heuristic methods, depends on the user's decision, in comparison to this methods with heuristic methods, meta-heuristic methods have many parameters that must be obtained by the user and experimentally. Therefore, the algorithms cannot obtain the same answers in similar iterations. In other words, these algorithms do not have a fixed routine to reach the solution and random parameters play a big role in these algorithms. For example, among these algorithms, we can refer to the methods of adaptive lage neighborhood algorithm [15], memetic algorithms[16], genetic reactive bone route algorithm [17], ant colony optimization [18], etc [19].
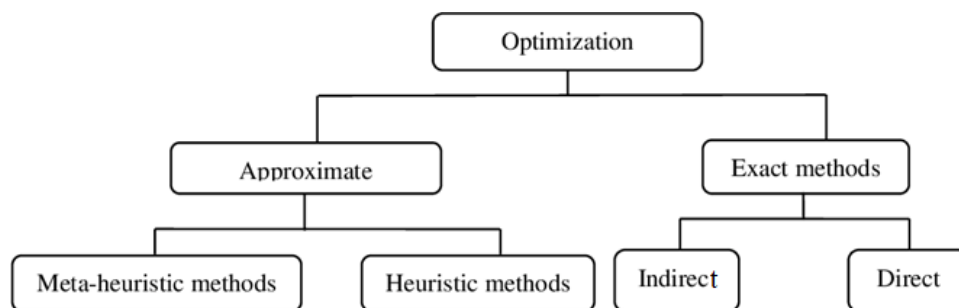

Figure 2. Classifications of the algorithms for solving the VRPs

The BCVP problem was first presented by Yousefikhoshbakht et al. in 2016 [20], where fairness between drivers for working time or revenue is considered. In other words, in this problem, taking into account a parameter such as alpha provided by the user, all the

distances covered should not differ from each other in terms of alpha percentage. This limitation causes that almost every driver's mission completion time to deliver or receive goods does not differ from each other by a certain amount. Imperialist competitive Algorithm (ICA) is one of the newest meta-heuristic algorithms, which has been used mostly on various continuous combinatorial optimization problems and has obtained very good results. On the other hand, due to the fact that the BVRP problem is of great importance in operational research issues, the purpose of this article is to address this problem and propose a hybrid ICA (HICA) algorithm for it for the first time. In the proposed method for the absorption stage, the optimization nearest neighbor algorithm is used, and in the revolution stage, the insert, swap and 2-opt improvement method are used. This algorithm has a very good efficiency compared to other meta-heuristic methods that will be discussed in the next sections.

In this article, Section 2 explains the imperialist competitive algorithm and the proposed method. Then, the computational results of the algorithm implemented on standard examples of transportation problems are analyzed in Section 3 and compared with meta-heuristic algorithms. Finally, the conclusion and future directions are presented in Section 4.

2. THE PROPOSED ALGORITHM

Optimization problems are so large in their practical dimensions that either exact optimization methods cannot easily achieve global optimal solution, or their optimal solutions cannot be achieved in an acceptable time because, due to their nature, they usually have multiple local optimas, and the existing optimization methods to solve these problems are very expensive and sometimes impossible. Nevertheless, these issues must be solved, and therefore there is no solution that is sufficient for near-optimal solutions in a way that has an acceptable quality and can be obtained in an acceptable period of time. Among the most important and promising effective methods that do not have the problems of complete counting methods are meta-heuristic algorithms that have recently been more focused on them. These methods have similarities with social and natural systems, and their use is derived from continuous heuristic methods that have had very good results in solving complex problems. These types of methods can provide a close to optimal in an acceptable time for a problem and are mostly based on counting methods, with the difference that they use additional information to guide the search. Also, meta-heuristic methods are completely general in terms of application and can solve complex compound problems. ICA algorithm is one of the newest meta-heuristic algorithms for solving optimization problems. This algorithm, like other evolutionary optimization methods, started with a number of initial population, and in this algorithm, each element of the population is called a country. The countries are also divided into two categories: colonies and colonizers, and each colonizer, depending on his power, dominates and controls a number of colonial countries. In addition, colonial countries try to depend on the colonial country as well as changing its culture and customs. Therefore, the colonized country should move in the direction of cultural assimilation with the colonizing country. It should be noted that in presenting the algorithm, this policy is carried out by moving the colonies of an empire according to a specific formula, which causes the goal function value of the colony country to improve if possible.

If during the movement of a colony, it reaches a better position compared to the colonizer, the place of the two will be exchanged; meanwhile, the total power of an empire is also defined as the total power of the colonizing country plus a percentage of the average power of its colonies. On the other hand, during the colonial competition, the weak empires gradually lose their power and disappear over time as they become weaker, and the colonial competition leads to a situation where there is only one empire that dominates the world. This situation happens when the colonial competition algorithm stops after reaching the optimal point of the objective function. Figure 3 shows the flowchart of the colonial competition algorithm with its stages.
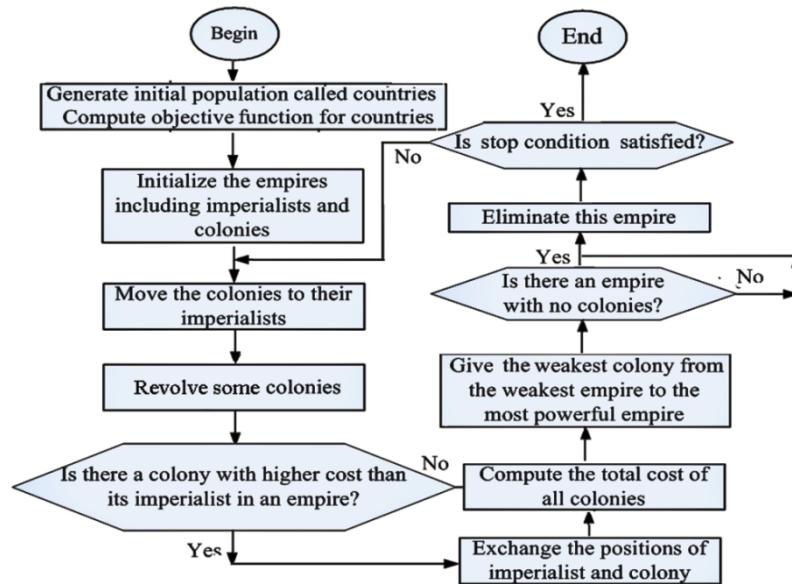
Figure 3. Flowchart of the classic ICA

Since the ICA method is an evolutionary method, it must be started by an initial population. Therefore, according to the number of countries in the world, N, an initial random population is created for each problem, and the values of the objective function of the VRP problem, whose value is equal to several Hamilton cycle, are obtained for each country. Now, the initial m countries that have better objective function values for the relevant problem are considered as colonizing countries, and by shifting, they form the indices 1 to m of the initial population. Now, because each colonizer has a colony equal to his power, therefore, the number of each colony for each colonizer is obtained from Eq (1),

$$k_j = \text{int}(\frac{1/f_j}{\sum_{i=1}^{m}(1/f_i)}.(N-m)) \quad j=1,...,m \tag{3}$$

It should be noted that at the end of these calculations $\sum_{j=1}^{m} k_j = N - m$, otherwise the rest of the remaining countries that did not belong to the colonizer will belong to the strongest colonizer. Now, having these initial hypotheses for the colonists, the algorithm starts working and continues until a stopping condition is reached. According to the existing

natural state, the colonial countries should move towards the colonial countries in terms of culture and society. In other words, in a two-dimensional problem, the initial population of the colonized country moves towards the colonizing country by the amount x and with an angle θ. In order to bring the colonized country closer to the colonizing country, it is used to select two points in the colonial country and arrange the members between them based on the colonizing country. For example, in this method, if [1 3 5 2 4|2 3] represents the colonial country and [1 4 2 3 5|3 2] represents the colonizing country for the BVRP with 5 nodes and two vehicles, then two random numbers between 2 and n (here 5 is) is considered (for example, suppose the number 2 and 4 are selected). Now, levels 2 to 4 are selected in the colonized country, which are 3, 5, and 2, respectively, and their order is obtained in the colonizing country, which is 2, 3, and 5, respectively. Therefore, these numbers replace the previous order and [1 2 3 5 4|2 3] is created. It should be noted that the obtained answer will replace the colonial country's answer only if it has a better objective function value. It is important to pay attention to the fact that in the colonial competition algorithm, since all the colonial countries perform the process of assimilation or the same combination only with the colonizing country, it is possible that countries similar to each other are formed, which causes the diversity of the population to disappear and the algorithm has premature convergence. To prevent this possibility, the tabu list is used for each empire. Therefore, a vector of length l is considered and both random numbers selected above are placed in this vector. Now this work continues until this list is filled. Like the tabu search algorithm, two new random numbers in the tabu list are accepted only if they lead to a better solution to the problem. It should be noted that when this list is filled, each new member replaces the oldest member. In other words, this list acts like a queue, and in the same way, this action causes the algorithm to prevent similar answers as much as possible.

In addition, in some cases, as it was said, the colonial countries can also become powerful and make a revolution. The meaning of the revolution is that the colonial countries grow regardless of the colonizing countries. For this purpose, in the proposed algorithm, three movements of insert, swap and 2-opt improver are used for the revolution operation. Figure 4 (left) shows the movement of the insert. In this method, a node is randomly selected and separated from its location, and then it is inserted in the best possible location that reduces the amount of traffic. In the swap movement shown in Figure 4 (middle), two nodes are selected and they change their place in the path. It should be noted that this move is accepted when it causes the cost of the tour to decrease compared to the previous state. The 2-opt improvement move shown in Figure 4 (right) works based on removing two non-neighboring edges from the net and reconnecting those two edges in another way. It should be noted that there are several ways to connect the path and create the tour again, but only a mode is accepted that applies to the limitations of the problem and the new tour obtains a better value for the problem. It is necessary to pay attention to the fact that the act of removing two edges and connecting them again continues consecutively so that no more double improvement movements are found. To pay attention to the fact that the probability of performing triple insert, swap, and 2-opt movements in the proposed algorithm is considered to be 40, 40, and 20%, respectively, because the two methods of insertion and displacement are weak local search movements that cause changes a little bit in the movement of the algorithm, but in the search movement of the 2-opt improver, it is a strong movement and it changes the direction of the algorithm suddenly to find better

answers, so it is better that the probability of doing this movement is lower than the previous two movements.
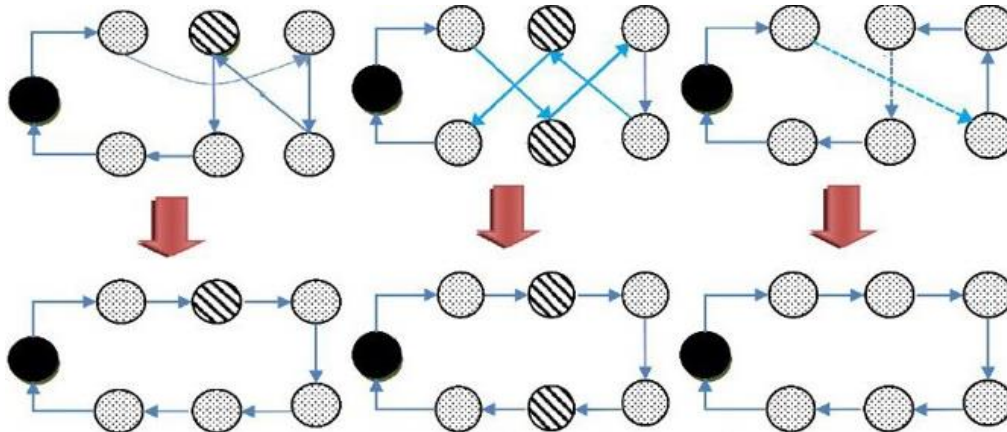

Figure 4. Insert, swap and 2-opt algorithms respectively

After the value of the objective function is obtained for all the colonial countries, it is possible that these countries have a better objective function than their colonizing countries. Therefore, the best colonial countries in each group are selected based on the objective function, and if they have a better objective function, their place will change, and so-called colonizing countries will emerge. It should be noted that if several colonial countries have the same and better objective function than their group, then one of them is randomly selected and compared with the colonizing country of its group. It is necessary to pay attention to the fact that there are two variables in the implementation of the algorithm that store the values of the best solution and the value of its objective function. These two variables become active after the colonizers appear. At this stage, the best answer and the objective function value of the colonizing countries are selected. Now, if the obtained value is better than the previous value obtained in the previous iterations, that answer and value of the objective function will replace the previous answer and value. In the next step, the power of each empire should be measured in order to determine which group has more power. For this purpose, Eq. (2) is used: These two variables become active after the colonizers appear.

$$c_j = f_j + \lambda(s_j) \qquad j = 1,...,m \qquad (2)$$

In this formula, c_j the total power of each empire, including the colonizing country and the countries colonized by that country, s_j is the average of the objective function of the colonial countries in each empire, *landa* is a multiplier between zero and one, which means that the influence of the objective function of the colonizing country compared to the average of the objective function of the colonial countries can be expressed as a power of one. Now that the power of the empires is determined, the weaker empire must begin to decline and lose its power by losing its colonies. Therefore, the weakest country is selected from the weakest empire and joins another empire. It should be noted that this annexation is not always assigned to the best empire, but based on the following probability for the *jth*

empire (Eq. (3)):

$$p_j = \frac{1/c_j}{\sum_{j=1}^{m} 1/c_j} \qquad j = 1,...,m \qquad (3)$$

Therefore, any empire that has more power is more likely to take over this country's colony. It should be noted that this condition is examined here, whether there is an empire that has no more colonies left for it or not. If an empire is found at this stage, that empire is removed, and otherwise the algorithm is repeated (each colony moves to its colonizing country) until the end condition of the loop is established. For the final condition of the loop, two conditions are used, which are examined simultaneously at the end of each iteration of the algorithm. These two conditions are repeating the algorithm for a certain number of t times and removing all the empires and leaving only one empire. If any of these two conditions happen at any time, the algorithm ends and the best answer and value that has been obtained so far is introduced as the best answer and value. Figure 5 shows steps of the proposed algorithm in more details.

> *1. Find N initial solutions to the vehicle routing problem and find the objective function value for all of them.*
> *2. Consider m to the initial population that have better objective function values and consider them as colonizers.*
> *3. According to the Eq. (1), assign the rest of the remaining countries to the colonists randomly and form m empires considering each colonizer and their colony.*
> *4. If a country from Eq. (1) cannot be attributed to empire, assign it to colonialism with the best objective function value.*
> *5. Bring each colony closer to the corresponding colonizer using the attraction function described for the problem (consider the tabu list).*
> *6. For a percentage of the colonies, consider the functions of insert, swap and 2-opt improvement and perform the revolution action on them based on the said percentage.*
> *7. If a better objective function value is obtained for a colony, replace it with the corresponding colonizer.*
> *8. Consider the best answer and value obtained in the entire population and update the best value obtained so far.*
> *9. Find the weakest colony of the weakest empire and assign it to one of the stronger empires based on Eq. (3).*
> *10. If an empire is found without a colony, remove it.*
> *11. If the repetition of the algorithm reaches a certain number or there is only one empire left, go to 12. Otherwise, go to 3.*
> *12. Print the best answer and algorithm value.*

Figure 5. Pseudo code of the proposed algorithm

3. COMPUTATIONAL RESULTS

In this part, first, the optimal combination of the parameters of the proposed hybrid ICA is obtained, and then the calculation results of this algorithm are compared with other meta-heuristic algorithms. All the codes of the proposed algorithm are written in MATLAB 7

language and the computer on which this program is implemented is Pentium 4 type with 3GHz power and eight gigabyte of memory. The optimal combination of parameters for the combined algorithm of colonial competition is shown in Table 1. In this table, five examples of table 2 have been considered and 5 tests have been done for each of them. Then according to the obtained values, the best ones are shown in the third column of Table 1. In this table, n represents the number of nodes in each example.

Table 1. The best values for the proposed algorithm's parameters

Variables	Candidate values	The best solutions
N	200,400,600,800,1000	800
m	N/10, N/20, N/30, N/40, N/50	N/20
λ	0.1, 0.2, 0.3, 0.4, 0.5	0.2
t	n, n/2, n/3, n/4, n/5	n
l	5,10,15,20,25	20

In order to be able to evaluate the effectiveness of the proposed algorithm for BVRP problems, it is necessary to present the number of standard problems for this category of problems and obtain the results by considering 20% as the maximum difference of the routes traveled by vehicles. It should be noted that in the first category, which uses the examples provided by Christofides [21], eight examples are considered that have the number of nodes between 10 and 45 and the number of vehicles that can be used for them is between 3 and 5. In these examples, whose full characteristics are presented in Table 2, the warehouse is considered to be approximately in the geographical center of the customers, and the customers are randomly scattered inside a square. In addition, for vehicles, the maximum amount of cargo loading and also for some of them, the maximum distance that can be traveled is considered. All distances apply in the Euclidean conditions of the problem and the number of vehicles is assumed to be a constant and homogeneous amount. In Table 2, the results of the proposed HICA algorithm are compared with three algorithms, in which two algorithms, rank ant system (RAS) and effective rank ant system (ERAS) from reference [20] are considered, and the third algorithm is an exact algorithm in the AIMMS software. This table consists of two parts. In the first part, for each of the examples, each algorithm has been executed ten times and the best solution obtained has been reported, while in the second part, the best solution has been considered and its *Gap* compared to the best known solution (*BKS*) has been reported. If *ans* is the best answer in each iteration, then *Gap* is obtained from the following formula.

$$Gap=100*(ans\text{-}BKS)/BKS \qquad (4)$$

It should be noted that in this table, *n* is the number of customers, *m* is the number of vehicles, *Q* is the capacity of each vehicle, *D* is the maximum distance that can be traveled for each vehicle, *Time* is second, *EA* is exact algorithm, and *BKT* is the best execution time for the best BKS. The results show that the RAS and ERAS algorithms were able to achieve the 3 and 5 best solutions, respectively, while despite the fact that this problem is NP-hard,

the 5 best solutions were found by the exact algorithm. Finally, it can be concluded that the proposed algorithm was able to achieve the best solutions in all eight examples. Of course, this performance for the proposed algorithm has been associated with the shortest execution times, so that each of these answers has been obtained in appropriate execution times. To see the average answers obtained for this example, see Figure 5, where the horizontal axis represents the names of the algorithms, while the vertical axis presents the mean value of the answers for these eight examples.

Table 2.Computational results for the proposed algorithm compared to other algorithms

Instance	n	m	Q	D	RAS	Time (RAS)	ERAS	Time (ERAS)	EA	Time (EA)	HICA	Time (HICA)	BKS	BKT
D1	10	3	120	∞	89.53	2.61	89.53	3.41	89.53	3	89.53	2.2	89.53	2.2
D2	15	5	100	∞	99.93	3.92	99.93	4.82	99.93	1452	99.93	3.7	99.93	3.7
D3	20	3	130	∞	121.62	5.01	121.62	6.29	121.62	6432	121.62	4.1	121.62	4.1
D4	25	4	120	∞	169.31	22.61	168.78	21.62	168.78	6582	168.78	15.21	168.78	15.21
D5	30	4	100	∞	256.6	41.90	249.66	34.81	249.66	3198	249.66	33.24	249.66	33.24
D6	35	4	120	200	295.53	91.71	295.53	85.88	298.51	20435	281.23	75.65	281.23	75.65
D7	40	5	120	160	349.21	95.45	335.63	94.82	341.80	98401	321.62	74.54	321.62	74.54
D8	45	5	120	200	423.18	99.61	396.41	99.71	499.51	243691	371.92	77.17	371.92	77.17
Instance	n	m	Q	D	RAS	Time (RAS)	ERAS	Time (ERAS)	EA	Time (EA)	HICA	Time (HICA)	BKS	BKT
D1	10	3	120	∞	0	18.64	0	55	0	36.36	0	0	89.53	2.2
D2	15	5	100	∞	0	5.95	0	30.27	0	39143.24	0	0	99.93	3.7
D3	20	3	130	∞	0	22.20	0	53.41	0	156778	0	0	121.62	4.1
D4	25	4	120	∞	0.31	48.65	0	42.14	0	43174.16	0	0	168.78	15.21
D5	30	4	100	∞	2.78	26.05	0	4.72	0	9520.94	0	0	249.66	33.24
D6	35	4	120	200	5.08	21.23	5.08	13.52	6.14	26912.56	0	0	281.23	75.65
D7	40	5	120	160	8.58	28.05	4.36	27.21	6.27	131911	0	0	321.62	74.54
D8	45	5	120	200	13.78	29.08	6.58	29.21	34.31	315684.6	0	0	371.92	77.17

If in the BVRP problem, the condition of balancing the paths is removed, then the VRP problem is obtained, which is one of the most important NP-hard problems, so it can be concluded that the BVRP problem is also an NP-hard problem. Now, although exact algorithms can be used to solve this problem in small-sized examples, when the size of the problem grows in industry and services problems, this type of algorithms can no longer be used, and meta-heuristic algorithms are one of the best methods to solve this problem [22-25]. For this purpose, to test the efficiency of the proposed algorithm, another group of examples is considered as the first group, and their full characteristics are presented in Table 3. This set contains between 50 to199 nodes as well as the depot and can be downloaded from the library website's following address:

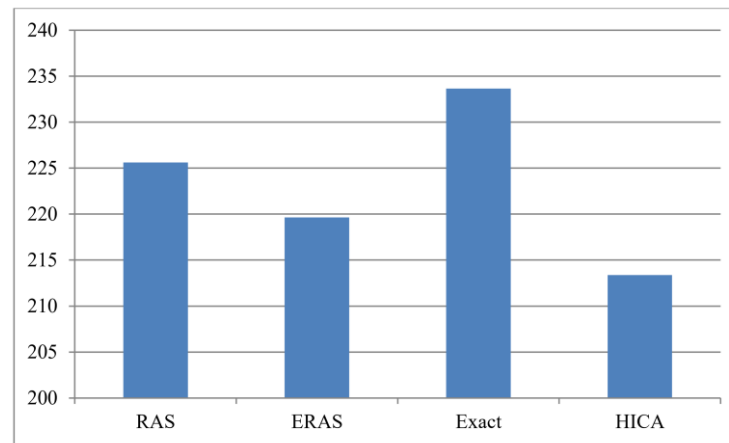URL: http://mscmga.ms.ic.ac.uk/jeb/orlib/vrpinfo.html

Figure 6. Means results of the algorithms

It should be noted that this class of examples has 5 to 18 vehicles. In addition, the results of the presented algorithm have been compared with the RAS and ERAS algorithms taken from reference [20]. The results shown in this table emphasize that the proposed algorithm is able to provide high-quality solutions for these problems and obtain the highest number of best solutions among them. In other words, the proposed algorithm has been able to achieve the best solutions obtained so far in all thirteen examples except for example C7.

Table 3.Computational results for the Christofides's benchmark

Instance	n	M	Q	D	RAS	ERAS	HICA	BKS
C1	50	5	160	∞	549.21	545.33	545.33	545.33
C2	75	10	140	∞	892.41	884.43	880.42	880.42
C3	100	8	200	∞	899.21	862.87	851.90	851.90
C4	150	12	200	∞	1285.08	1134.54	1083.82	1083.82
C5	199	17	200	∞	1592.33	1402.21	1304.82	1304.82
C6	50	6	160	200	598.03	598.03	598.03	598.03
C7	75	11	140	160	989.65	965.31	973.11	965.31
C8	100	9	200	230	934.81	934.81	934.81	934.81
C9	150	14	200	200	1311.93	1264.61	1203.87	1203.87
C10	199	18	200	200	1583.76	1461.72	1406.72	1406.72
C11	120	7	200	∞	1274.69	1201.42	1158.72	1158.72
C12	100	10	200	∞	921.49	904.37	889.56	889.56
C13	120	11	200	720	1784.86	1721.80	1702.62	1702.62
C14	100	11	200	1040	939.41	931.81	931.81	931.81

5. CONCLUSION

In this article, a modification of the imperialist competitive method was presented, in which three local improvement methods were used to improve the obtained answers. In addition to the fact that the proposed algorithm has a higher quality than other meta-heuristic algorithms for solving the BVRP problem. Due to the relatively new nature of the algorithm, many works can be suggested to further increase its efficiency, including the combination of the

corresponding algorithm with other meta-heuristic methods, the use of more efficient methods for the movement of the colonized countries towards the colonizing countries, the use of stronger local algorithms such as Lin-Kernigan, etc. On the other hand, this algorithm can be used for other combination optimization problems such as the vehicle routing problem with time windows and its extensions. Working on these ideas and making them practical will be postponed to the next articles.

REFERENCES

1. YousefiKhoshbakht M, Mahmoodi Darani N. A combined metaheuristic algorithm for the vehicle routing problem and its open version, *J AI Data Min*, 2019, 7(1): 169-179.
2. Yousefikhoshbakht M, Malekzadeh N, Sedighpour M. Solving the traveling salesman problem based on the genetic reactive bone route algorithm with ant colony system. *Int J Produc Man Eng*, 2016, **4**(2): 65-73.
3. Maleki F, Yousefikhoshbakht M. A hybrid algorithm for the open vehicle routing problem. *Int J Opt Civ Eng*, 2019. **9**(2): 355-71.
4. Guo Q, Wang N. The Vehicle Routing Problem with Simultaneous Pickup and Delivery Considering the Total Number of Collected Goods. Mathematics, 2023, **11**(2): 311.
5. Yousefikhoshbakht M, Didehvar F, Rahmati R. Solving the Heterogeneous Fixed Fleet Open Vehicle Routing Problem by a Combined Meta-heuristic Algorithm, *Int j product res*, 2014, **52**(9): 2565-2575.
6. Yousefikhoshbakht M, Dolatnejad A. A column generation for the heterogeneous fixed fleet open vehicle routing problem. *Int J Product Man Eng*, 2017, **5**(2): 55-71.
7. Contardo C, Desaulniers G, Lessard F, Reaching the elementary lower bound in the vehicle routing problem with time windows. Networks, 2015, **65**: 88–99.
8. Ahmed ZH, Yousefikhoshbakht M. An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows. *Alex Eng J*, 2023, **64**: 349-63.
9. Zhang X, Chen L, Gendreau M, Langevin, A. Learning-based branch-and-price algorithms for the vehicle routing problem with time windows and two-dimensional loading constraints. *INFORMS J Comput*, 2022, **34**(3): 1419-1436.
10. Fisher M. Optimal solution of vehicle routing problems using minimum K-trees, *Oper Res*, 1994, **42** (4): 626–642.
11. Zhang X, Chen L, Gendreau M, Langevin A. A branch-and-cut algorithm for the vehicle routing problem with two-dimensional loading constraints. *Eur J Oper Res*, 2022, **302**(1): 259-269.
12. Christofides N, Mingozzi A, Toth P. The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. Combinatorial Optimization. Chichester: Wiley; 1979, 315-38.
13. Yellow P. A computational modification to the saving method of vehicle scheduling. *Oper Res Quar*, 1970, **21**: 281-283.
14. Yousefikhoshbakht M, Sedighpour M. An optimization algorithm for the capacitated vehicle routing problem based on ant colony system. *Aust J Bas Appl Sci*, 2011, **5**(12): 2729-2737.

15. Wen M, Sun W, Yu Y, Tang J, Ikou K. An adaptive large neighborhood search for the larger-scale multi depot green vehicle routing problem with time windows. *J Clean Product*, 2022, **374**: 133916.
16. Zhou Y, Kou Y, Zhou M. Bilevel memetic search approach to the soft-clustered vehicle routing problem. *Trans Sci*, 2023, **57**(3): 701-16.
17. 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 Product Man Eng*, 2016, **4**(2): 65-73.
18. 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.
19. Ahmed, ZH, Yousefikhoshbakht M. A Hybrid Algorithm for the Heterogeneous Fixed Fleet Open Vehicle Routing Problem with Time Windows. *Symmetry*, **15**(2), 486, 2023.
20. Yousefikhoshbakht M, Didehvar F, Rahmati F. An effective rank based ant system algorithm for solving the balanced vehicle routing problem, *Int J Ind Eng*, 2016, **23**(1).
21. Christofides N, Mingozzi A, Toth P. The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. Combinatorial Optimization. Chichester: Wiley; 1979, 315–38.
22. Kaveh A, Talatahari S. A charged system search with a fly to boundary method for discrete optimum design of truss structures, *Asia J Civ Eng*, 2010, **11**(3): 277-293.
23. Kaveh A, Malakoutirad S. Hybrid genetic algorithm and particle swarm optimization for the force method-based simultaneous analysis and design, *Iran J Sci Tech*, 2010, **34**: 15-34.
24. Kaveh A, Talatahari S. Hybrid charged system search and particle swarm optimization for engineering design problems, *Eng Comput*, 2011, **28**: 423-440.
25. Kaveh A, Zakian P. Enhanced bat algorithm for optimal design of skeletal structures, *Asia J Civ Eng*, 2014, **15**: 179-212.