DOI: 10.24996/ijs.2022.63.10.21

ISSN: 0067-2904

The Best Efficient Solutions for Multi-Criteria Travelling Salesman Problem Using Local Search Methods

Manal Ghassan Ahmed*, Faez Hassan Ali
Mathematics Department, College of Science, Mustansiriyah University, Baghdad, Iraq

Received: 19/9/2021 Accepted: 29/11/2021 Published: 30/10/2022

Abstract
In this research, we propose to use two local search methods (LSM's); Particle Swarm Optimization (PSO) and the Bees Algorithm (BA) to solve Multi-Criteria Travelling Salesman Problem (MCTSP) to obtain the best efficient solutions. The generating process of the population of the proposed LSM's may be randomly obtained or by adding some initial solutions obtained from some efficient heuristic methods. The obtained solutions of the PSO and BA are compared with the solutions of the exact methods (complete enumeration and branch and bound methods) and some heuristic methods. The results proved the efficiency of PSO and BA methods for a large number of nodes (n). The proposed LSM's give the best efficient solutions for the MCTSP for $n \leq 700$ jobs in a reasonable time.

Keywords: Travelling Salesman Problem, Local Search Method, Particle Swarm Optimization, Bees Algorithm.

افضل الحلول الكفوءة لمسالة البائع المتجول المتعددة المعايير بأستخدام طرق البحث المحلية

منال غسان احمد*, فائز حسن علي
قسم الرياضيات , كلية العلوم , الجامعة المستنصرية , بغداد , العراق

الخلاصة
في هذا البحث اقترحنا طريقتين من طرق البحث المحلية (LSM's) هما أمثلية سرب الجسيمات (PSO) و خوارزمية النحل (BA) لحل مسألة البائع المتجول المتعددة المعايير (MCTSP) ولإيجاد افضل الحلول الكفوءة لها. تم اعتماد عملية توليد مجتمع الطريقتين على التوليد العشوائي او من خلال اضافة حلول ابتدائية مولدة من بعض الطرق التقريبية الكفوءة. كذلك تم مقارنة الحلول المستحصلة من طرق LSM's مع نتائج الطرق الدقيقة المضبوطة (مثل طريقتي العد التام و التفرع والتقييد) والطرق التقريبية. النتائج اثبتت كفاءة PSO وBA ولعدد كبير من العقد (n). الطرق المحلية المقترحة اعطت حلول كفوءة لحل مسالة MCTSP لـ $n \leq 700$ وفي وقت مقبول.

1. Introduction
The Travelling Salesman Problem (TSP) is the problem of finding the minimum expensive to visit a set of cities, a particular sequence begins and ends at the same city, and each city

*Email: manalghassan@uomustansiriyah.edu.iq

must be exactly visited one time. Since this problem was formulated mathematically, the essence of the problem was in the area of combinatorial optimization. There is an important difference that can be made between the symmetric TSP and the asymmetric TSP. For the symmetric case, all distances are equal ($d_{ij} = d_{ji}$) no matter what it was. If we travel from city (i) to city (j) or on the contrary because the distance is the same. In the second case, the distances are not equal for all pairs of cities. These kinds of problems arise when we do not transact with locative distances between cities but with the time and cost associated with travelling between locations [1].

The complexity of TSP is considered NP-Complete Problem. When the number of cities increases, the complexity of TSP will increase exponentially [2]. TSP is formulated in intensive things and it is considered an optimization problem.

Most applications of TSP are logistic, manufacture of microchips, planning and DNA sequencing, so that customers, DNA fragments or soldering points have represented a city in TSP, while the times, cost, or measure between DNA fragments are considered distance. Also, one of the important applications of TSP is in astronomy [2].

There are many studies of Local search methods (LSM's) on TSP, Meng et al. [3] proposed a new method based on a discrete Artificial Bee Colony algorithm for TSP. They redefined the searching strategy and transforming mechanism of leading bees, following bees and scout bees according to discrete variables. Finally, the experimental results show that the new algorithm can relatively find a satisfactory solution in a short time, and improve the efficiency of solving the TSP.

Ding [4] proposed a multiobjective by using one of the most important optimization solving methods like the Particle Swarm Optimization (PSO) method. This method depends upon improved culture, because the standard PSO algorithm has a slow convergence speed and is easy to fall into the local minimum problem. In order to solve these problems, the suggested algorithm has high convergence with minimum local value.
Myszkowski et al.[5] suggested a new method to solve multiobjective TSP (they notation it by mTSP) where this method is better than other methods which is called Sorting Genetic Algorithm II (NSGA-II). The new modification considers the improvment of the results which are verified by the benchmark of (mTSP).

Jasim and Ali in 2019 [6], one of the used exact methods is Branch and Bound Technique (BABT), they proposed to use Simulated Annealing (SA) and Genetic Algorithm (GA). They proposed two more LSMs which are called improved GA (IGA) and Hybrid GA (HGA) to enhance the results of SA and GA. HGA proved its efficiency compared with other methods for $n \leq 2000$. In addition, they used the successive rule to reduce the size of the problem. Finally, they applied the suggested methods in a practical example which they called the Iraqi Cities Problem (ICP).

This research consists of the following sections: In section 2, the formulation of the Multicriteria Travelling Salesman Problem (MCTSP) is presented. In section 3, some LSMs are discussed. In section 4, the results of the LSMs, which are used in this paper, are compared with some exact and heuristic methods. Finally, in section 5, a discussion and analysis of the results are introduced.

2. Formulation of the Multi-Criteria Travelling Salesman Problem (MCTSP)

In this research, we devoted to MCTSP which is the distance and the time that traveller salesman is taken from the one city to another city. The problem can be described as follows: There are n number of cities and a salesman must visit each city once only and return to the original starting city. Where:
d_{ij}: is the distance from city i to city j.
t_{ij}: is the time from city i to city j.
We derive the following mathematical model for MCTSP as follows:
Minimize $Z = (F, G)$
Where $F = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij}$ and $G = \sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij}x_{ij}$
Subject to:

$$\left.\begin{aligned} &\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,2,\dots,n. \\ &\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,2,\dots,n. \\ &x_{ij} \in \{0,1\} \end{aligned}\right\} \quad \dots(P)$$

The objective function of the P-problem is to minimize F and G simultaneously which can be written in matrix notation as follows: Let $D = [d_{ij}]$ be the distance matrix and $T = [t_{ij}]$ be the time matrix, while $X = [x_{ij}]$, $i, j = 1,2,\dots,n,$ is the adjacency matrix of the graph of the P-problem.

Now, let π be a vector, sequence, permutation or solution where $\pi \in S, S$ is the set of all feasible solutions to the P-problem. Suppose we start with and end at city 1, then $\pi = \{P_1, P_2, \dots, P_{n-1}\}$, s.t. $P_j \in \{2,3,\dots,n\}$ then the cost can be written as:

$$Z_C = c_{1,\pi(1)} + \sum_{i=1}^{n-2} c_{\pi(i),\pi(i+1)} + c_{n-1,1} \qquad \dots(1)$$

Where Z_C is the cost of distance ($Z_d = F$) or cost of time ($Z_t = G$), and $c_{ij} = d_{ij}$ or t_{ij}. Relation (1) grantees that the salesman will return to the city 1.

Then the objective function of the P-problem can be stated as follows:

$$\text{Minimize } Z = (F, G) \qquad \dots(2)$$

Where $F = d_{1,\pi(1)} + \sum_{i=1}^{n-2} d_{\pi(i),\pi(i+1)} + d_{n-1,1}$ and $G = t_{1,\pi(1)} + \sum_{i=1}^{n-2} t_{\pi(i),\pi(i+1)} + t_{n-1,1}$.

From relation (2) we can introduce a COST_MCTSP algorithm which is designed to calculate the MCTSP objective function: the COST_MCTSP is as follows:

<u>**COST_MCTSP(π)**</u>
Step(1): INPUT: $n, D = [d_{ij}]$, $T = [t_{ij}]$, $i, j = 1,2,\dots,n$.
Step(2): $F(D_\pi) = D(1, \pi(1)) + D(\pi(n), 1)$;
Step(3): FOR $j = 1{:}\,n-1$
  $F(D_\pi) = F(D_\pi) + D(\pi(j), \pi(j+1))$;
ENDFOR $\{j\}$
Step(4): $G(T_\pi) = T(1, \pi(1)) + T(\pi(n), 1)$;
Step(5): FOR $j = 1{:}\,n-1$
  $G(T_\pi) = G(T_\pi) + T(\pi(j), \pi(j+1))$;
ENDFOR $\{j\}$
Step(6): **OUTPUT**: $Z(F(D_\pi), G(T_\pi))$;
Step(7): END.

3. Local Search Methods (LSMs)

In this section, we used two LSMs, these LSMs are PSO and Bees Algorithm (BA) to find the best efficient solutions for solving MCTSP.

In general, the LSMs are very suitable for solving Combinatorial Optimization Problem (COP), especially for large of cities. These methods depend on generating natural individuals. The crossover between these individuals is a considerable concept for swarm intelligence. These individuals can communicate with each other in many ways to improve the new individual [7].

3.1 Particle Swarm Optimization (PSO)

Kennedy and Eberhard in 1985 are the first researchers in artificial intelligence to create Particle Swarm Optimization (PSO) [4]. PSO aims to optimize the objective function by changing the information of the swarm. Every particle or individual can be considered a solution. Those solutions know the best solution which is called a global best solution then they will improve their position and velocity according to the global solution. This information is the objective function of each solution. From these solutions, we will search for the best solution depending on its position and velocity in the swarm [7].

The following two relations are the basic concepts:

$$v_{id} = w * v_{id} + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id}) \quad \ldots(3.a)$$

$$x_{id} = x_{id} + v_{id} \quad \ldots(3.b)$$

Where

w is the inertia weight for convergence,

c_1 and c_2 are positive constants,

r_1 and r_2 are random functions in the range [0,1],

$X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ represents the i^{th} particle;

$P_i = (p_{i1}, p_{i2}, \ldots, p_{id})$ represents the (pbest) best previous position (the position gives the best fitness value) of the i^{th} particle; the symbol g represents the index of the best particle among all the particles in the population,

$V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ represents the rate of the position change (velocity) for particle i [8].

3.2 Bees Algorithm (BA)

BA is a search algorithm that uses honeybees' food foraging strategy to find the best solution to a given optimization problem. Any point in the search space is regarded as a potential food source. 'Scout bees' randomly sample the space (i.e. solutions are created at random) and, using the fitness function, find the best solution (i.e. the solutions are assessed). Other 'bees' are assigned to check the fitness landscape in the area of the highest-ranking sites, and the sampled solutions are ranked. A 'flower patch' is the field surrounding a solution [9].

4. Comparison Results for MCTSP

In a small town, let $d_{ij} \in [1,10]$ in km, if we consider that the maximum velocity $v = 60\ km/h$, then:

$$t_{ij} = \begin{cases} 1,2,\ldots,10, & \text{if } d_{ij} = 1,2,3, \\ 2,3,\ldots,10, & \text{if } d_{ij} = 4,5,6, \\ 3,4,\ldots,10, & \text{if } d_{ij} = 7,8,9, \\ 4,5,\ldots,10, & \text{otherwise.} \end{cases}$$

in minutes.

The d_{ij} and t_{ij} values, for all examples, are generated randomly and uniformly.

In this paper, we run the suggested LSMs as follows: We use 100 iterations for $n = 5, \ldots, 10$,

use 500 iterations for $n = 11, \dots, 40$ and use 1000 iterations for $n = 60, \dots, 700$.
The two discussed local search methods (LSMs) have population size (20) solutions as an initial population.

Before applying the LSMs and obtaining the results in tables, we introduce some important abbreviations:

n	:	Number of cities
Av	:	Average for five examples.
$ANES$	:	Average number of efficient solutions for (5) examples.
AT	:	Average of CPU-Time per second for (5) examples.
F	:	Objective Function of P-problem.
Z_{op}	:	Optimal objective function.
Z_{ap}	:	Approximation objective function.
AE	:	Absolute Error $\lvert Z_1 - Z_2 \rvert$.
RAE	:	Relative Absolute Error $\lvert Z_1 - Z_2 \rvert / Z_2$.
R	:	0 < Real < 1.
LSM(0)	:	The 1st, 2nd and the set of r $(r \geq 1)$ solutions of the initial population generated by MDTM, MTDM and MVGM respectively [10], while the remaining solutions are obtained randomly.
LSM(1)	:	The initial population of LSM all generated randomly.

It is important to mention that the methods:

CEM	:	Complete Enumeration Method,
MDTM	:	Minimizing Distance-Time method,
MTDM	:	Minimizing Time-Distance method,
BAB	:	Branch and Bound,
MVGM	:	Multi-Variable Greedy Method.

are obtained from reference [10].

The comparison results between PSO(0), PSO(1), BA(0) and BA(1), for P-problem, for $n = 10, 30, 60, 80, 100, 300, 500, 700$ are shown in Table).

Table 1- Comparison between PSO(0), PSO(1), BA(0) and BA(1), $n = 10{:}\,700$.

n	PSO(0)			PSO(1)			BA(0)			BA(1)		
	***Av*(Z_{ap})**	***AT***	***ANES***	***Av*(Z_{ap})**	***AT***	***ANES***	***Av*(Z_{ap})**	***AT***	***ANES***	***Av*(Z_{ap})**	***AT***	***ANES***
10	(32.7,40.6)	*R*	6.0	(33.9,40.0)	*R*	4.4	(33.3,40.2)	*R*	6.2	(39.1,42.0)	*R*	4.2
30	(72.0,101.2)	1.0	5.0	(117.5,147.8)	*R*	7.4	(72.0,101.2)	*R*	5.6	(129.5,156.2)	*R*	7.0
60	(120.2,164.4)	3.4	5.0	(263.8,299.0)	2.6	6.2	(120.2,164.4)	1.0	5.8	(280.0,313.6)	*R*	8.6
80	(150.5,192.9)	4.1	4.4	(364.1,418.7)	3.5	8.8	(150.5,192.9)	1.5	4.4	(367.8,427.1)	*R*	6.8
100	(174.9,238.5)	5.8	4.8	(466.4,536.4)	4.3	6.0	(174.9,238.5)	2.6	4.8	(473.4,546.6)	*R*	6.8
300	(382.7,456.1)	178.6	2.6	(1536.1,1689.1)	13.1	7.4	(382.7,456.1)	177.2	2.6	(1521.7,1726.6)	2.5	9.0
500	(588.1,645.1)	579.8	2.2	(2566.0,2879.3)	21.9	6.4	(588.1,645.1)	576.2	2.2	(2574.6,2905.8)	3.9	8.0
700	(796.1,849.4)	991.9	2.0	(3615.7,4108.7)	30.3	7.2	(796.1,849.4)	844.9	2.0	(3653.0,4100.4)	5.8	9.0
Av.	**(289.7,336.0)**	**220.6**	**4.0**	**(1120.4,1264.9)**	**0.5**	**6.7**	**(289.7,336.3)**	**200.4**	**4.0**	**(1129.9,1277.3)**	**1.5**	**7.4**
RAE	-----	---	---	**(2.87,2.76)**	--	---	----------	--	----	**(2.90,2.80)**	--	----

Note: In Table 1, since the results of PSO(0) and BA(0) are better than PSO(1) and BA(1), so we depend on LSM(0) in the comparison results with other methods.

Table 2: Comparison between PSO(0) and BA(0) with CEM, for $n = 5:12$.

n	CEM			PSO(0)			BA(0)		
	Av(Z_{op})	***AT***	***ANES***	***Av***(Z_{ap})	***AT***	***ANES***	***Av***(Z_{ap})	***AT***	***ANES***
5	(19.2,21.3)	*R*	2.0	(19.2,21.3)	*R*	2.0	(19.2,21.3)	*R*	2.0
6	(24.8,30.2)	*R*	4.4	(24.8,30.2)	*R*	4.0	(24.8,30.2)	*R*	4.0
7	(23.2,26.3)	*R*	4.4	(23.4,25.9)	*R*	4.0	(23.4,26.9)	*R*	4.0
8	(32.2,35.4)	*R*	7.2	(30.9,35.3)	*R*	5.0	(32.0,36.8)	*R*	5.0
9	(30.8,35.9)	*R*	6.8	(33.5,37.5)	*R*	6.0	(35.3,38.0)	*R*	5.0
10	(31.1,39.6)	*R*	8.4	(32.7,40.6)	*R*	6.0	(33.2,42.6)	*R*	4.0
11	(30.2,40.5)	8.2	8.2	(31.6,40.2)	*R*	5.0	(36.1,48.1)	*R*	5.0
12	(32.4,43.6)	104.6	9.6	(36.5,47.1)	*R*	5.0	(39.6,47.9)	*R*	4.0
Av.	**(28.0,34.1)**	**14.1**	**6.7**	**(29.1,34.8)**	***R***	**4.6**	**(30.5,36.5)**	***R***	**4.1**
AE	-----	---	---	**(1.1,0.7)**	--	---	**(2.5,2.4)**	--	--

Note: In Table 2, for $n = 8$, we notice that the $Av(Z_{ap})$ of PSO(0) and BA(0) are better than CEM, of course that impossible, the reason is that the set of efficient solutions are different and the number of ANES is different too for each method.

Table 3: Comparison results between PSO(0) and BA(0) with BAB, $n = 15:5:40$.

n	BAB			PSO(0)			BA(0)		
	Av(Z_{op})	***AT***	***ANES***	***Av***(Z_{ap})	***AT***	***ANES***	***Av***(Z_{ap})	***AT***	***ANES***
15	(37.7,46.2)	6.1	10.0	(40.9,58.9)	*R*	3.0	(41.1,57.4)	*R*	2.0
20	(42.7,56.9)	18.2	9.2	(56.8,67.3)	*R*	3.0	(56.8,67.3)	*R*	3.0
25	(50.6,67.7)	173.1	13.8	(61.1,85.2)	*R*	5.0	(61.1,85.2)	*R*	5.0
30	(57.0,76.3)	226.7	13.4	(72.0,101.2)	1.0	5.0	(72.0,101.2)	*R*	5.0
35	(64.4,78.9)	516.8	17.4	(80.0,109.9)	1.2	5.0	(78.6,109.8)	*R*	5.0
40	(67.8,86.9)	1206.7	18.2	(89.1,118.2)	1.4	4.0	(89.1,118.2)	*R*	4.0
Av.	**(53.4,68.8)**	**357.9**	**13.6**	**(66.7,90.1)**	**0.6**	**4.1**	**(66.5,89.9)**	***R***	**4.0**
AE	------	---	---	**(13.3,21.3)**	--	---	**(13.1,21.1)**	--	---

In Figure 1, we will summarize the comparison results between PSO(0) and BA(0) with CEM and BAB (see Tables 1 and 2) for *P*-problem, for $n = 5:5:40$.

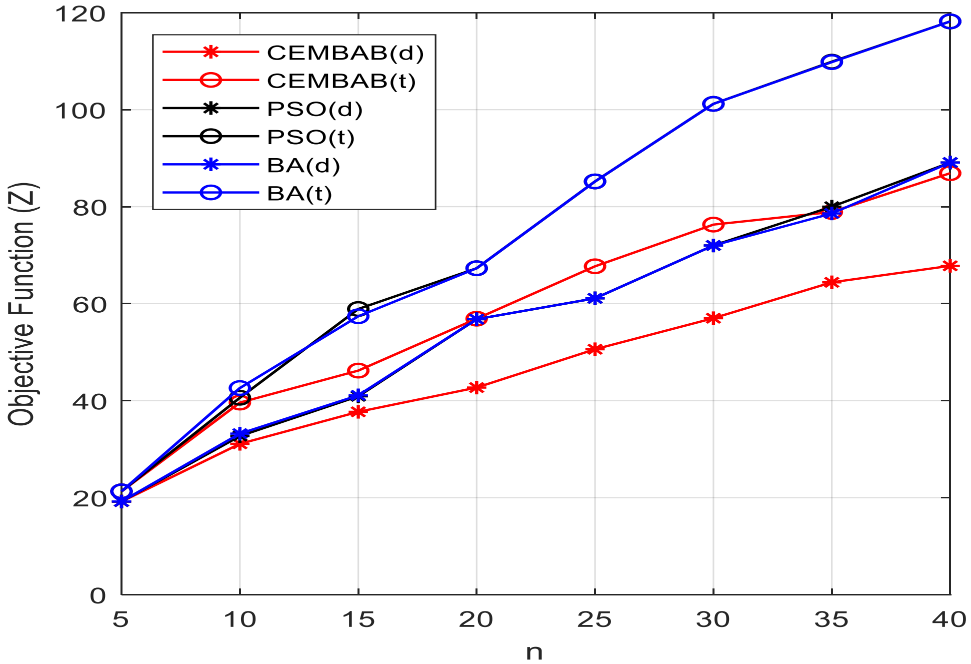

Figure 1: The results comparison between PSO(0) and BA(0) with CEM and BAB, $n = 5:5:40$.

Table 4: Comparison results between PSO(0) and BA(0) with MVGM, $n = 60,80,100,300,500,700$.

n	MVGM			PSO(0)			BA(0)		
	Av(Z_{ap})	***AT***	***ANES***	***Av***(Z_{ap})	***AT***	***ANES***	***Av***(Z_{ap})	***AT***	***ANES***
60	(122.5,168.7)	*R*	4.6	(120.2,164.4)	3.4	5.0	(120.2,164.4)	1.0	5.0
80	(158.9,211.8)	*R*	3.6	(150.5,192.9)	4.1	4.0	(150.5,192.9)	1.5	4.0
100	(201.6,260.2)	2.2	3.6	(174.9,238.5)	5.8	4.0	(174.9,238.5)	2.6	4.0
300	(428.1,549.0)	223.8	1.4	(382.7,456.1)	178.6	2.0	(382.7,456.1)	177.2	2.0
500	(875.5,1033.0)	766.4	1.8	(588.1,645.1)	579.8	2.0	(588.1,645.1)	576.2	2.0
700	(913.0,1036.4)	1251.0	2.6	(796.1,849.4)	991.9	2.0	(796.1,849.4)	844.9	2.0
Av.	**(449.9,543.2)**	**373.9**	**2.9**	**(368.8,424.4)**	**293.9**	**3.1**	**(368.8,424.4)**	**267.2**	**3.1**
AE	--------	----	----	**(81.1,118.8)**	----	----	**(81.1,118.8)**	----	----

In Figure 2, we will show the comparison results between PSO(0) and BA(0) with MVGM (see Table 4) for P-problem, for $n = 60,80,100,300,500,700$.

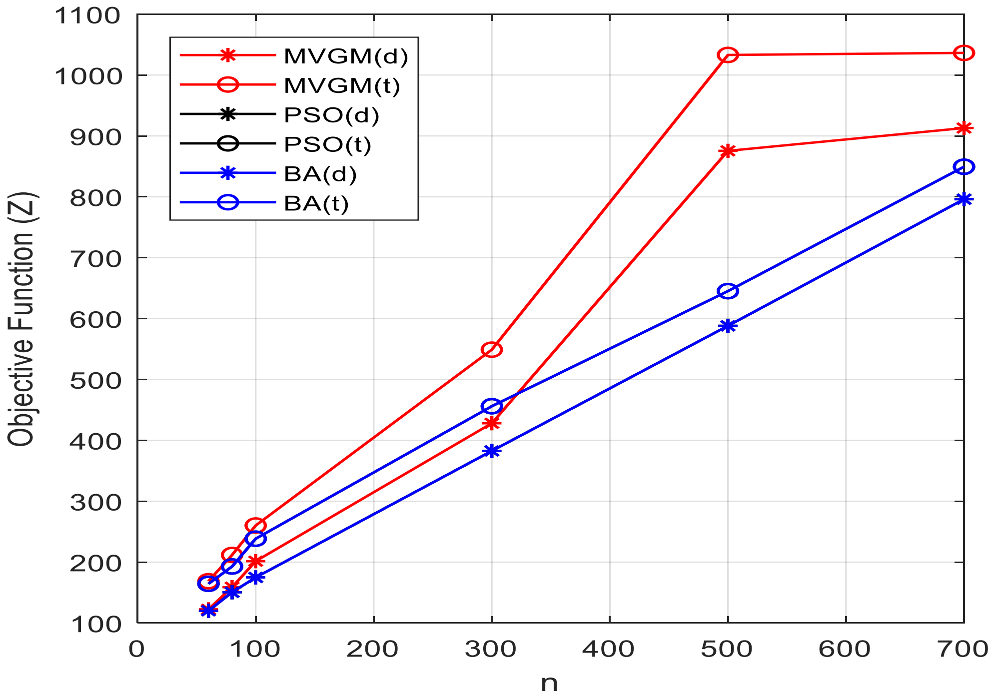

Figure 2: the comparison results between PSO(0) and BA(0) with MVGM, for $n = 60,80,100,300,500,700$.

5. Discussion and Analysis of Results

In this section, we discuss and analyse the results that we obtained in this paper as follows:

- From the comparison results, which are shown in Table 1, for the P-problem, of PSO(0), PSO(1), BA(0) and BA(1), for $n = 10{:}700$, we notice that PSO(0) and BA(0) are more efficient than PSO(1) and BA(1), respectively. Therefore, we used only the results of PSO(0) and BA(0) in the comparison results with other methods.
- We compare the results between PSO(0) and BA(0) with CEM for P-problem, for $n = 5{:}12$. We notice that PSO(0) is close to CEM than BA(0) (see Table 2).
- From the comparison results between PSO(0) and BA(0) with the BAB method for P-problem, for $n = 15{:}5{:}40$, we notice that PSO(0) and BA(0) are closed to each other's and close to BAB, but PSO(0) takes a long time than BA(0) (see Table 3).
- From the comparison results between PSO(0) and BA(0) with a heuristic method MVGM for P-problem, for $n = 60,80,100,300,500,700$. We notice that PSO(0) and BA(0) are efficient and better than MVGM (see Table 4).

6. Conclusions

1. In this study, asymmetric multi-criteria TSP (i.e., $d_{ij} \neq d_{ji}, \forall i, j$) which is an NP-hard problem is studied.
2. We proposed two LSM; PSO and BA to find the set of the best efficient solutions (tours) in a reasonable time.
3. We conclude that for PSO and BA, it is better to start with good initial solutions to decrease the consuming time and number of iterations to obtain efficient results for P-problem.
4. PSO is better than BA $= 5{:}12$. While, BA is better for $n = 15{:}5{:}40$, while for $n \geq 50$, the two LSMs give very close results.

5. For the consuming time, we conclude that BA is better than PSO in obtaining final results for *P*-problem.
6. For *P*-problem, the results of PSO and BA are better than the heuristic MVGM up to n = 700.
7. From the topics of interest to us in the future, we recommended the researchers should use more LSMs (like simulated annealing, genetic algorithm,…, etc.) to obtain good efficient and approximation solutions for *P*-problem for $n \gg 100$.
8. In future work, we suggest to studying the weighted multi-objective functions TSP to find the exact and approximate solutions for the problem.
9. For future work, we suggest making a hybrid between PSO and BA to exploit the good performance of each method to obtain more improved results.
10. Since the LSMs; PSO and BA are efficient in giving good results of MCTSP, so we have to use them in solving life practical applications (like the Iraqi Cites Problem).

References

[1] Kaml B. S and Ibrahim M. S., "Solving the Multi-Objective Travelling Salesman Problem with Real Data Application, "*Journal of Al-Nahrain University,* vol.21, no. 3, September, pp. 146-161, 2018. DOI: 10.22401/JNUS.21.3.18.

[2] Ejim, S. "Implementation of Greedy Algorithm in travel Salesman Problem," Reg. No:12/28213/D/1, Abubakar Tafawa Balewa University, Bauchi, September 2016.

[3] Meng L., Yin S. and Hu X., "A New Method Used for Traveling Salesman Problem Based on Discrete Artificial Bee Colony Algorithm, " Telkomnika, vol.14, no.1, pp. 342-348. March 2016.

[4] Zhiping D., "Research of improved particle swarm optimization algorithm, " *AIP Conference Proceedings* 1839, 020148 (2017).

[5] Myszkowski P. B., Laszczyk M. and Dziadek K., "Non–dominated Sorting Tournament Genetic Algorithm for Multi-Objective Travelling Salesman Problem, " *Proceedings of the Federated Conference on Computer Science and Information Systems,* vol. 18, pp. 67–76, 2019. DOI: 10.15439/2019F192, ISSN 2300-5963 ACSIS,

[6] Jasim S M., Ali F. H., "Exact and Local Search Methods for Solving Travelling Salesman Problem with Practical Application, " *Iraqi Journal of Science,* vol. 60, no.5, pp. 1138-1153, 2019.

[7] Cergibozan C., "Metaheuristic Solution Approaches for Traveling Salesman and Traveling Repairman Problems," thesis, IZMIR, 2013.

[8] Shi Y., "Particle Swarm Optimization", Electronic Data Systems, Inc. Kokomo, IN 46902, USA Feature Article, IEEE Neural Networks Society, February, (2004).

[9] D. T. Pham and M. Castellani, "The Bees Algorithm: modeling foraging behavior to solve continuous optimization problems", Anniversary 50th Issue paper, Manufacturing Engineering Centre, Cardiff University, Cardiff, UK, DOI: 10.1243/09544062 JMES1494, 2009.

[10] Ahmed M. Gh. and Ali F. H., "New Exact and Heuristic Methods for Solving Multi-Criteria Travelling Salesman Problem, " *6th National Scientific/ Third International Conference, College of Education for Pure Sciences / Kerbala University, Iraq, AIP Conference Proceedings* (ISSN: 0094-243X). 6th April, 2021(Accepted).