IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 6, Ser. I (Nov - Dec 2019), PP 60-65
www.iosrjournals.org

Applications of Ion Motion Optimization Algorithm in the Travelling Salesman Problem

Thi Tuyen Nguyen[1], Thi Thu Dao[1]
[1]*Department of Computer Science*
University of Information and Communication TechnologyThai Nguyen, Vietnam
Corresponding Author: Thi Tuyen Nguyen

Abstract: *The Travelling Salesman Problem is a classic problem used to evaluate the effectiveness of optimization algorithms. Ion motion optimization algorithm is a newly published optimization algorithm. This algorithm was inspired by the movement of ions in nature and It is attracting the attention of many studies in the world. This paper aims to evaluate the effectiveness of IOM in Travelling Salesman Problem. It is compared to some metaheuristic search algorithms in the same class such as Genetic Algorithm, Particle Swarm optimization, Ant Colony Optimization. The results show that the Ion Motion Optimization algorithm uses fewer parameters, has a better convergence speed, faster than the rest of the algorithms.*

Date of Submission: 04-12-2019 Date of Acceptance: 19-12-2019

I. Introduction

The Traveling Salesman Problem (TSP) is a problem that has many practical applications. It is of the NP-hard type and this problem is often used as the standard problem to evaluate the effectiveness of new algorithms that are used to solve combinatorial optimization problems [7].

The TSP problem comes from the fact that a salesperson wants to find the shortest journey from his city to go through all cities to sell products and finally return to the starting city. The important thing here is that each city can only be passed exactly once.

Mathematically, the TSP problem is a problem of finding the shortest Hamiltonian path on a weighted complete graphs $G = (N, A)$, where N is a set of vertices corresponding to a set of cities, A is the set of edges that connect cities whose weight is the corresponding length. Note that if the graph is not complete, we can always add the missing edges to get a new full graph G ' and the weight of these edges is large enough for the optimal path on G' to be also optimizer on G. We denote the length of each side is $dij(i,j)\in A$) corresponding to the distance between the i-th city and the j-th city (for all i, $j\in N$).

Because of its common nature, TSP is often chosen as the problem to evaluate the effectiveness of optimal algorithms. So far, there have been many ways to solve this problem. Among them are some meta-heuristic algorithms such as Genetic Algorithm (GA) [11], [13], Particle Swarm optimization (PSO) [14], [15], Ant Colony Optimization (ACO) [8], [9], [10].

Ions Motion Optimization (IMO) is an optimization algorithm published by Javidy in 2015 [1]. This is an algorithm inspired by the motion of ions in nature. The results of this algorithm have been evaluated by the author on 10 benchmark functions. These results have attracted a series of studies on applying IMO to specific problems [2], [3], [4]. However, there has been no assessment of the application of IMO to the TSP problem. Therefore, in the next section of this article, we will discuss the IMO algorithm in more detail. Next, we evaluate the application of the IMO algorithm for the TSP problem. The results were compared with the same class algorithms such as GA genetic algorithm, PSO herd optimization, ACO ant group optimization.

II. The Imo Algorithm for The Tsp Problem

A. *The IMO algorithm*

The term ion is a Greek term. Michael Faraday, a famous English physicist, introduced the term in 1834. Generally, charged particles are called ions and can be divided into two types (anions and cations).

The conceptual model of anions and cations is illustrated in Figure 1. The main inspiration of the IMO algorithm is the fact that ions of similar electrical charge tend to repel, while ions of opposite charge attract each other.

The solutions to a certain optimization problem using the IMO algorithm are divided into two groups. These are anions (negative ions) and cations (positive ions). The ions represent the candidate solution to a specific problem and the attraction / repulsion moves the ions around the search space.

In the IMO algorithm, each ion will move to the best ions that have an opposite charge to it. Ions are evaluated based on their fitness. Therefore, the fitness of the ions is directly proportional to the value of the object function. Obviously, the anion moves towards the best cation, while the cation moves towards the best anion. The amount of their motion depends on the attraction / repulsion between them.

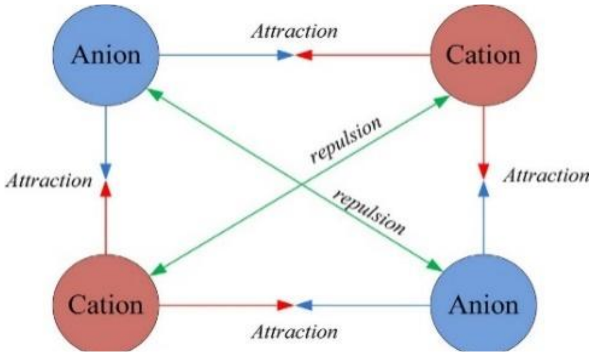

Figure 1. Conceptual and anion models, cations, attraction and repulsion [1].

The movement of ions in the IMO algorithm can ensure the diversification and enhancement of search algorithms. This is based on two completely different phases: liquid phase versus solid phase.

- **Liquid phase**

The ions in the liquid move more freely. In addition, attraction forces between ions with opposite charge are more than repulsion forces between ions with the same sign. In this phase, IMO ignores the thrust to explore the search space.

- **Solid phase**

In the solid phase, the ions were concentrated at the optimum point and the convergence occurred. However, due to the indefinite shape of the search space, convergence may occur for local optimization. Therefore, this phase provides a mechanism for the ions to escape local optimum.
Figure 2 illustrates the distribution of ions in the solid phase. This phase maintains the diversity of ions during optimization. After this stage, the ions again enter the liquid phase and explore the search space.

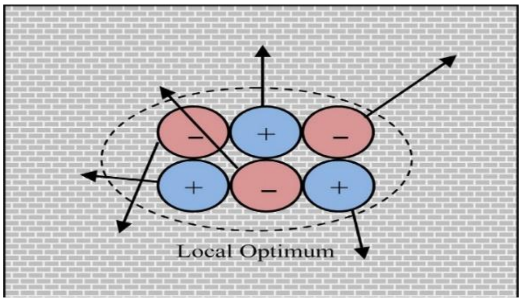

Figure 2. Ions move around optimum in the solid phase [1]

B. *IMO algorithm for TSP problem*

The general steps of the IMO algorithm for the problem are shown in Figure 3. Specifically:

Step 1. Initializing the IMO population

- Create parameters of IMO algorithm such as the number of iterations M, random values.
- Randomize initial population including N / 2 Anion and N / 2 Cation. Each individual in the initial population is a N-dimensional vector containing N vertices of the TSP problem. Thus, each individual is the permutation of the vertices to be passed in the TSP problem

Step 2 Evaluating fitness value

- Fitness value of each individual is the total distance when performing a TSP cycle
- Update the best and worst Anions and Cations

Step 3. Position updating of Ion in liquid phase

- Calculate distances of anions to the best Cation and distances of cations to the best Anions (1)

$$AD_{i,j} = d(A_{i,j}, Cbest_j) \qquad CD_{i,j} = d(C_{i,j}, Abest_j) \tag{1}$$

- Determine attraction forces between j-th component of the i-th anion to the best cation, determine attraction forces between the j-th component of the i-th cation to the best anion (2).

$$AF_{i,j} = \frac{1}{1+e^{-0.1/AD_{i,j}}} \qquad CF_{i,j} = \frac{1}{1+e^{-0.1/CD_{i,j}}} \tag{2}$$

- After calculating the force, the position of anion and cation is updated as follows (3):

$$\begin{array}{ll}
A_{i,j} = A_{i,j} + ceil(AF_{i,j} \times d(Cbest_j, A_{i,j})) & C_{i,j} = C_{i,j} + ceil(CF_{i,j} \times d(Abest_j, C_{i,j})) \\
for\ m = 0 \text{ to } j-1 & for\ m = 0 \text{ to } j-1 \\
if\ A_{i,j} \neq A_{i,m}\ and\ A_{i,j} < N & if\ C_{i,j} \neq C_{i,m}\ and\ C_{i,j} < N \\
A_{i,j} = A_{i,j} & C_{i,j} = C_{i,j} \\
else\ Re-initialized\ A_{i,j} & else\ Re-initialized\ C_{i,j} \\
endif & endif \\
endfor & endfor
\end{array} \tag{3}$$

Step 4. Position updating of ions in solid phase (4)

$$\begin{array}{l}
if\,(CbestFit) >= CworstFit/2\ and\ AbestFit >= AworstFit/2 \\
\quad if\ rand() > 0.5 \\
\qquad A_i = \left(A_i + ceil(\phi_1 \times (Cbest - 1))\right) \bmod N \\
\quad else \\
\qquad A_i = \left(A_i + ceil(\phi_1 \times Cbest)\right) \bmod N \\
\quad endif \\
\quad if\ rand() > 0.5 \\
\qquad C_i = \left(C_i + ceil(\phi_2 \times (Abest - 1))\right) \bmod N \\
\quad else \\
\qquad C_i = \left(C_i + ceil(\phi_2 \times Abest)\right) \bmod N \\
\quad endif \\
\quad if\ rand() < 0.05 \\
\qquad Re-initialized\ A_i\ and\ C_i \\
\quad endif \\
endif
\end{array} \tag{4}$$

Step 5. Checking for repeat and stop conditions of IMO algorithm

- If the end condition is satisfied (eg maximum number of repetitions) the process is terminated, the result is the best ion among the Anions and Cations.
- If stop condition is not reached, repeat from step 2.

In (3), (4) the function *ceil ()* is used because the value of the j-th components in each ion is an integer. When we update the j-th component, this updated value is compared to the previous values (from 0 to j-1) so that there is no duplication. If duplication occurs, the j-th component is randomly recreated.

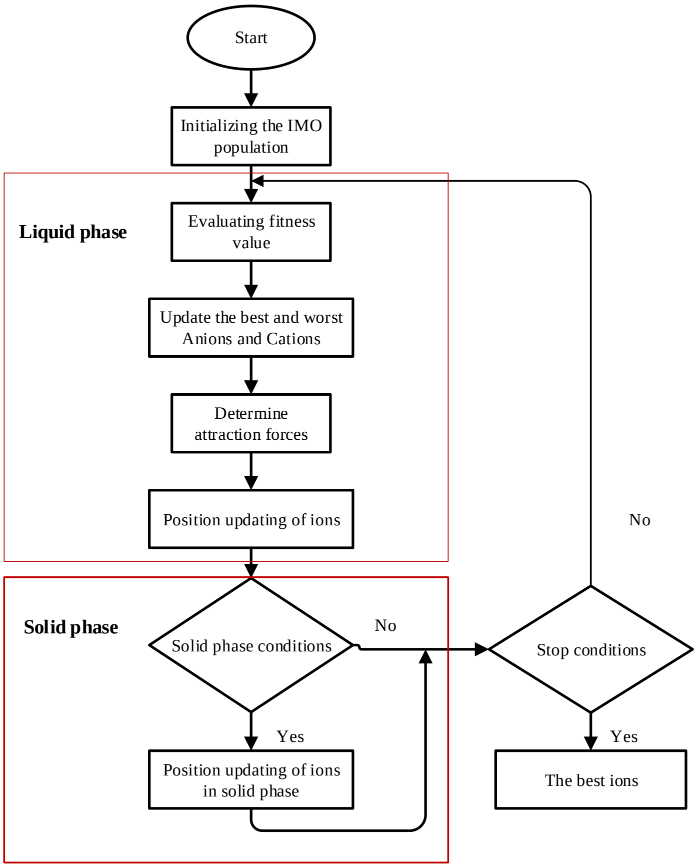

Figure 3. General steps of the IMO algorithm

III. Result Simulations And Results

Table 1. Installation parameters of algorithms

Parameters	GA	PSO	ACO	IMO
Population size	100	100	100	100
Maximum number of iterations	1,000	1,000	1,000	1,000
Maximum velocity	N/A	0.50	N/A	N/A
Learn factor	N/A	$c_1 = 2$ $c_2 = 2$	N/A	N/A
Crossover operator	Single point	N/A	N/A	N/A
Crossover rate	0.90	N/A	N/A	N/A
Mutation operator	Real value	N/A	N/A	N/A
Mutation rate	0.01	N/A	N/A	N/A
β	N/A	N/A	2.0	N/A
Evaporation coefficient	N/A	N/A	0.05	N/A
Q	N/A	N/A	100	N/A
q0	N/A	N/A	0.09	N/A
R0 for crossover strategy	N/A	N/A	0.33	N/A
$\phi 1$	N/A	N/A	N/A	0.5
$\phi 2$	N/A	N/A	N/A	0.5

To evaluate the effectiveness of the IMO algorithm for the TSP problem, we compare the performance of this algorithm with algorithms such as GA, PSO, ACO. The algorithms are evaluated on the same condition as the number of iterations, population size, TSP input data. Parameters of the above algorithms are shown in Table 1.

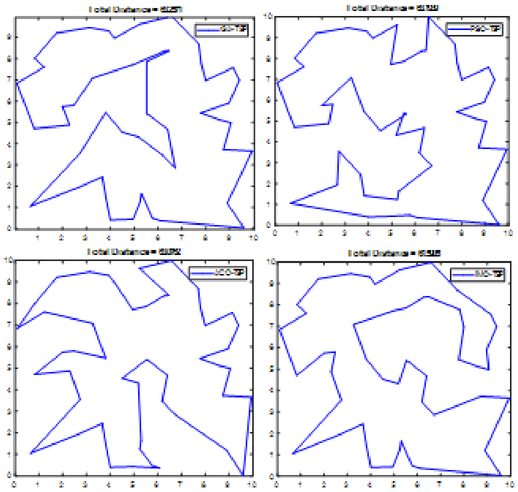

Figure 4. Simulation results of TSP problem with 50 cities

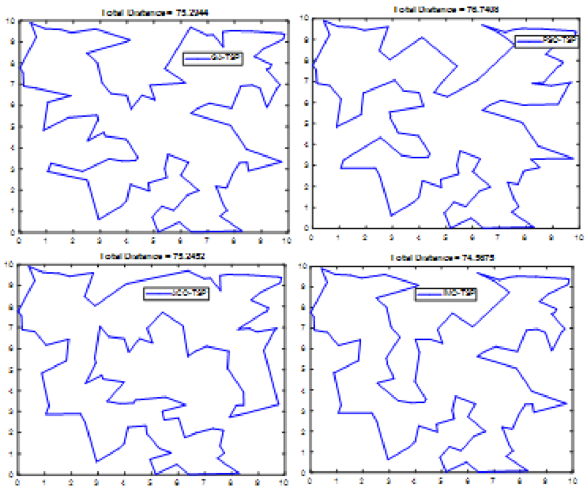

Figure 5. Simulation results of TSP problem with 100 cities

First, algorithms are implemented on random TSP data. This data is generated by users with 50 nodes that are distributed in two dimensions with a value of less than 10. The maximum number of iterations selected is 1000. The results are shown in Figure 4. IMO show the best results. The remaining algorithms have worse results, respectively ACO, GA, PSO. In general, it can be seen that with the number of iterations of 1000, the obtained results of the algorithms are acceptable (because the lines connecting vertices do not intersect). However, the result of the IMO algorithm is still close to the best solution.

Figure 5 illustrates the results obtained on a topology of 100 randomly distributed cities in the range [0.10]. The maximum number of iterations is now set to 10000. From the results obtained it can be seen that the algorithms both give acceptable solutions and the algorithm for the best results is still the IMO.

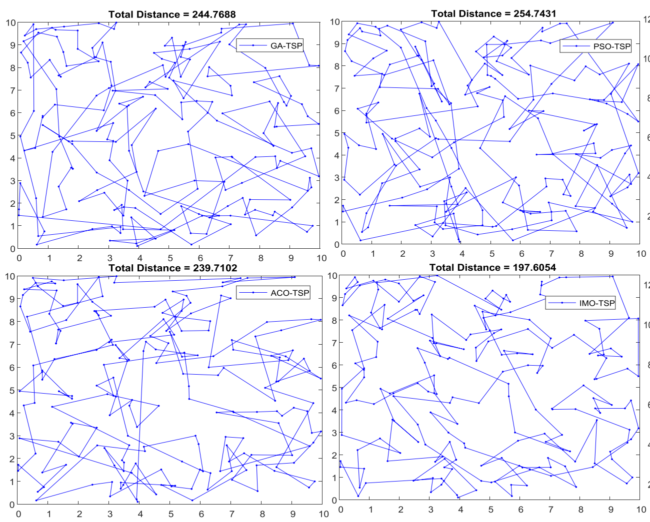

Figure 6. Simulation results of TSP problem with 200 cities

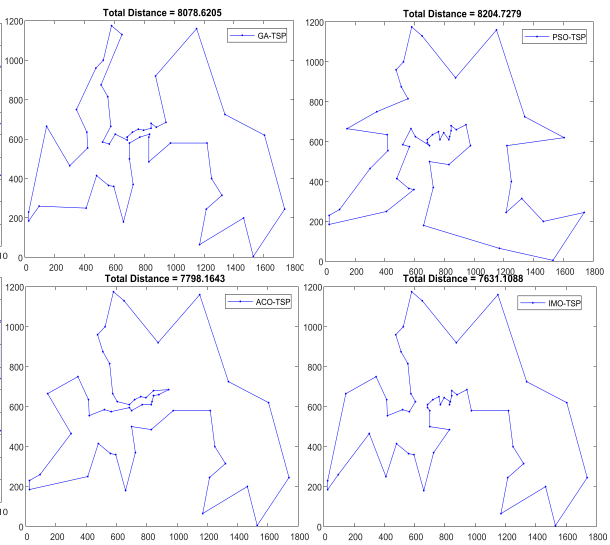

Figure 7. Simulation results of TSP problem with Berlin52

In Figure 6, the number of cities is increased to 200. Because 10000 iterations are not enough for the convergence of algorithms, so all four algorithms have not reached the optimal solution (lines that connect vertices still intersect). However, in this case, the IMO algorithm still gives better results than the other algorithms.

Table 2 The comparisons for Berlin52

Algorithm	GA	PSO	ACO	IMO
Scale	52	52	52	52
Optimal Solution	7,542	7,542	7,542	7,542
Best	8,078.6205	8,204.7279	7,798.1643	7,631.1088
Worst	8,243.02	8,589.31	8,080.57	7,718.31
Average	8,176.55	8,319.51	7,932.31	7,686.42
Time	134.23	24.56	160.52	15.26

In Figure 7, we evaluate the efficiency of four algorithms GA, ACO, PSO, IMO on the standard data sample Berlin52. This data sample is often selected by the international scientific community as test data to evaluate the effectiveness of optimal search algorithms. The maximum number of iterations here is still selected as 1000. The results show that IMO has the best solution compared to the remaining three algorithms. The detailed results are shown in Table 2.

IV. Conclusion

This paper proposed to apply IMO algorithm to TSP problem. By comparing the results obtained on random data and on standard sample data (recognized by the international scientific community), it is confirmed that IMO is an easy-to-install algorithm. This algorithm has the least number of parameters, the fastest convergence time and the most accurate results compared to the algorithms PSO, GA, ACO.

In fact, there are many problems that the IMO algorithm can be applied to find the optimal solution. Besides, we can not confirm that IMO has advantages for all combination optimization problems. However, the result of this algorithm for the TSP problem is a potential sign for the application of the IMO algorithm on other optimal search problems such as telecommunications network routing, finding the optimal path for robots, finding the optimal parameter for the controller.

Acknowledgment

This research was supported by the ICTU research grant (under project # T2019-07-05)
We thank our colleagues from the ICTU who provided insight and expertise that greatly assisted the research.

References

[1]. B. Javidy, A. Hatamlou, and S. Mirjalili, Ions motion algorithm for solving optimization problems, Applied Soft Computing, vol. 32, no. 3, pp. 72-79, 2015.

[2]. Cheng-Hong Yang, Kuo-Chuan Wu, Li-Yeh Chuang, Breast Cancer Risk Prediction Using Ions Motion Optimization Algorithm, Journal of Life Sciences and Technologies Vol. 4, No. 2, December 2016, pp. 49-55

[3]. Kumar, Mohit & Dhillon, J.S.. (2018). An Experimental Study of Ion Motion Optimization for Constraint Economic Load Dispatch Problem, IEEE international conference of power energy, environment and intelligent control.

[4]. Nguyen TT., Wang MJ., Pan JS., Dao T., Ngo TG. (2020) A Load Economic Dispatch Based on Ion Motion Optimization Algorithm. In: Pan JS., Li J., Tsai PW., Jain L. (eds) Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies, vol 157. Springer, Singapore

[5]. Sabry Ahmed Haroun, Benhra Jamal, El Hassani Hicham, A Performance Comparison of GA and ACO Applied to TSP, International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 19, pp. 28-35, May 2015

[6]. Goldbarg, Elizabeth & Goldbarg, Marco & Souza, Givanaldo. (2008). Particle Swarm Optimization Algorithm for the Traveling Salesman Problem. 10.5772/5580.

[7]. Htun, Thi. (2018). A Survey Review on Solving Algorithms for Travelling Salesman Problem (TSP). International Journal of Scientific and Research Publications (IJSRP). 8. 10.29322/IJSRP.8.12.2018.p8481.

[8]. Bifan Li, Lipo Wang & Wu Song “Ant Colony Optimization for the Traveling Salesman Problem Based On Ants with Memory” Fourth International Conference on Natural Computation (2009).

[9]. L. Li, S. Ju, Y. Zhang, Improved Ant Colony Optimization for the Traveling Salesman Problem, International Conference on Intelligent Computation Technology and Automation, IEEE, 2008.

[10]. Zar Chi Su SuHlaing and May Aye Khine, “Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm”, International Journal of Information and Education Technology, Vol. 1, No. 5, December 2011.

[11]. Dwivedi, TarunaChauhan, SanuSaxena and PrincieAgrawal, “Travelling Salesman Problem using Genetic Algorithm”, International Journal of Computer Applications (IJCA), 2012, pp. 25-30.

[12]. Kumar N, Karambir, Kumar R. “A Genetic algorithm approach to study Traveling salesman problem”. Journal of Global Research in Computer Science. 2012; (3):33–8.

[13]. Gupta S, Panwar P. “Solving Traveling salesman problem using genetic algorithm” International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE). 2013; 3(6):376–80.

[14]. X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu, Q.X. Wang,” Particle swarm optimization-based algorithms for TSP and generalized TSP”, Information Processing Letters 103 (2007) 169–176,2007.

[15]. Huilian FAN,” Discrete Particle Swarm Optimization for TSP based on Neighborhood”, Journal of Computational Information Systems 6:10 (2010) 3407-3414.

Thi Tuyen Nguyen. "Applications of Ion Motion Optimization Algorithm in the Travelling Salesman Problem." IOSR Journal of Computer Engineering (IOSR-JCE) 21.6 (2019): 60-65.