GELS-GA: Hybrid Metaheuristic Algorithm for Solving Multiple Travelling Salesman Problem

Ali A. R. Hosseinabadi[1], Maryam Kardgar[2], Mohammad Shojafar[3], *Student Member, IEEE*, Shahaboddin Shamshirband[4], *Student Member, IEEE*, and Ajith Abraham[5,6]

[1]*Young Research Club, Islamic Azad University, Behshahr Branch, Behshahr, Iran*
Email: a.r.hosseinabadi@iaubs.ac.ir
[2] *Department of Computer, Mirdamad University, Gorgan Branch, Gorgan, Iran*
Email: kardgar_karyam@yahoo.com
[3]*DIET department, University Sapienza of Rome, Eudossiana 18, P.O.Box 00184, Rome, Italy*
Email: shojafar@diet.uniroma1.it
[4]*Department of Computer System & Technology, University of Malaya, Kuala Lumpur, Malaysia*
Email: shahab1396@gmail.com
[5]*Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, USA*
[6]*IT4Innovations,VSB – Technical University of Ostrava, Czech Republic*
Email: ajith.abraham@ieee.org

***Abstract*— The Multiple Traveling Salesmen Problem (mTSP) is of the famous and classical problems of research in operations and is accounted as one of the most famous and widely used problems of combinational optimization. Most of the complex problems can be modeled as the mTSP and then be solved. The mTSP is a NP-Complete one; therefore, it is not possible to use the exact algorithms for solving it instead the heuristics methods are often applied for solving such problems. In this paper, a new hybrid algorithm, called *GELS-GA*, has been presented for solving the mTSP. The utility of *GELS-GA* is compared with some related works such as GA and ACO and achieves optimality even in highly complex scenarios. Although, the proposed algorithm is simple, it includes an appropriate time of completion and the least traversed distance among existing algorithms.**

Keywords- Multiple Traveling Salesmen Problem (mTSP); Optimization; Genetic Algorithm (GA); Gravitational Emulation Local Search (GELS) algorithm; Newton Law.

I. INTRODUCTION

Optimization problems, either single-objective or multi-objective, are generally said to be NP-hard and nobody can solve them efficiently by any known algorithm in a reasonable time. In fact, many problems that seem simple are difficult to solve because when the number of combinations increases exponentially, the size of problem increase. Searching for every possible combination is extremely computationally expansive and unrealistic. The most famous example in this field is the Traveling Salesman Problem (TSP) in which computational complexity increases exponentially when the number of cities increases [1]. And in mTSP a salesman visit a certain list of cities, and each city is visited exactly once, and return to his initial location, and total cost of travel must be minimized, and every salesman must start at one node and ends at the same node [2,3].

The TSP consists of finding the shortest closed route to visit all cities. Several heuristic methods have been proposed to solve it, Including Classical Search Maps, Simulated Annealing (SA), Artificial Neural Networks (NNs), Genetic Algorithms (GAs), Evolutionary Programming, Ant Colony Optimization (ACO), Tabu Search (TS), fine-tuned learning, etc. Although the TSP has received a great deal of attention, research on the mTSP is limited. Only exact algorithms, Heuristic algorithms, neural network-based methods, and ant systems have all been proposed to solve the mTSP.

mTSP is an extension of TSP, so it is a NP-hard problem too. It seems to be more appropriate than the TSP for practical applications and we can use it for solve the real world problems, such as railway transportation, routing and pipeline laying, and to simulate many everyday applications such as transportation logistics, job planning, vehicle scheduling, and so on [2,3]. In general, the mTSP is defined as follows: a set of nodes, that known as *m* salesmen and located at a single depot node. The other nodes (cities) that are to be visited are called intermediate nodes. Therefore, the mTSP consists of finding tours for all m salesmen, who all start and end at the depot, and each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. The cost metric can be distance, time, and so on [4]. Possible variations of the problem are as follows:

- **Single vs. multiple depots:** All salesmen start from and end their tours at a single point in the single depot case. But, in the multiple depote case there are several depots with a number of salesman located at each, and the salesmen can either return to their original depot after completing their tour or return to any depot but the initial number of salesmen at each depot must be remain the same after all the travel;
- **Number of salesmen:** The number of salesman in the problem may be a bounded variable or fixed a priori;
- **Fixed charges:** When the number of salesman is not fixed, so each salesman usually has an associated fixed cost. In this case, the minimization of the

number of salesman that are activate in the solution is a concern;

- **Time windows:** In this case, certain nodes must be visited in specific periods, called time windows. This is an important extension of the mTSP and known as the multiple traveling salesman problem with time windows (mTSPTW). This case has some applications in school bus, ship and airline scheduling problems [4].

Compared to the TSP, the mTSP is suitable to model real life situations, because it can handle more than one salesman. These situations arise from various routing and scheduling problems. Several exact solutions in mTSP procedures are include: branch-and-bound type methods [5, 6], which are suitable only for solve the problems of reasonable sizes. On the other hand, there are some heuristic techniques, of which Neural Network-based procedures [7, 8] are the most popular and also some approaches such as [9] explains the routing method in the network avoiding considering artificial models. There are also some transformation-based procedures [10, 11], which transformed the mTSP to the standard TSP, and do not seem efficient, since the resulting TSP is highly degenerate, especially with the increasing number of salesman [4].

In this paper, we proposed a novel combinatorial algorithm namely GELS-GA to solving the multiple traveling salesman problem. Then, performance of the GELS-GA has been compared with well-known optimization algorithms such as the GA and the ACO. Simulation results show superiority of the GELS-GA over the other existing optimization algorithms in the literature.

The rest of this paper is organized as follows: The following section provides a brief overview of the related work on solving the TSP and mTSP. In Section III, the joint of GELS and genetic approach (GA) is used to solve the symmetric mTSP. Simulation results are shown in Section IV and performance of the GELS-GA is compared to other existing optimization algorithms in the literature. Finally, a conclusion is given in Section V.

II. RELATED WORK

In the last two decades, the TSP received quite big attention, and various approaches have proposed to solve the problem, e.g. branch-and-bound [12], neural network [13] or Tabu search [14]. Some of these methods are exact algorithms, while others are near-optimal or approximate algorithms. In following some of these algorithms for solving, the mTSP was introduced.

Because of the fact that TSP is already a complex, namely an NP-complete problem, heuristic optimization algorithms, like genetic algorithms (GAs) need to take into account. The aim of in [15] a detailed overview was given about how genetic algorithms can be applied to solve these problems and a novel representation based genetic algorithm has been developed to the specific one depot version of mTSPTW. The main benefit is the transparency of the representation that allows the effective incorporation of heuristics and constrains and allows easy implementation.

Authors in [16] analyze the general properties of mTSP, and find that the multiple depots and closed paths in the graph is a big issue for mTSP. Then proposed a novel method Known as the MDCP to solve a heterogeneous mTSP with multiple depots and closed paths it. In their method, a complicated graph is transformed into a simplified one firstly, and then an effective algorithm is used to solve the mTSP based on the simplified results. In addition, an optimization method that is similar to 2-OPT, was proposed and used to optimize the general results. Optimization method can decrease the total cost of MDCP to a large extent [16]. Another approach to the well-known traveling salesman problem (TSP) using self-organizing maps known as MGSOM was introduced in [17], the purpose of authors is to look for the incorporation of an efficient initialization methods and the definition of a parameters adaptation law to achieve better results and a faster convergence. Like the existing heuristic, the modified heuristic possesses many of these advantages of a good heuristic for the TSP solution. These advantages are: (1) easy implementation, (2) fast computation, (3) robust applicability, and (4) production of good solutions. The MGSOM is well suited for larger instances of the TSP since it has a fast convergence and low complexity [17]. Authors in [18] designed an ant system to solve fixed destination multi-depot multiple traveling salesmen problem (MmTSP). Their method can solve new problems without any renewed tunings. Also this method can introduce answers very close to the optimal solution in a very short time compared to the exact approach used in Lingo 8.0 [18].

III. GELS-GA APPROACH

In this paper, a combination of GA and GELS algorithm has been used to solve mTSP. The objective of presenting the proposed algorithm is to combine the public search capabilities of GA with local search of GELS algorithm and to create a stable algorithm, which can make reaching the global optimum largely possible. In the proposed algorithm, one-dimensional array with length of all existing cities plus the number of salesmen has been used to display chromosome. Each cell of the chromosome called gene is indicative of each city's number from indexes zero to $n-1$ and each of genes are indicative of the number of cities, those that each traveling salesman will visit, from indexes n to the end of array.

Fig. 1 shows a chromosome with length of 13, which represents the number of cities and traveling salesmen. The number of cities begins from zero in this chromosome, for example the first, second and tenth homes respectively refers to the eighth, forth, and seventh cities. As you can see in the structure of chromosome shown in Fig. 1, the last three homes of array refer to the number of cities, which will be visited by each traveling salesman. For example, the first salesman will visit the first three cities including the eighth, forth and third; the second salesman will visit next three cities including the ninth, first and sixth; and finally the third salesman will visit the last four cities. In the proposed algorithm, each chromosome indicates the number of cities, their paths and the order of traveling cities by each one of the salesman.

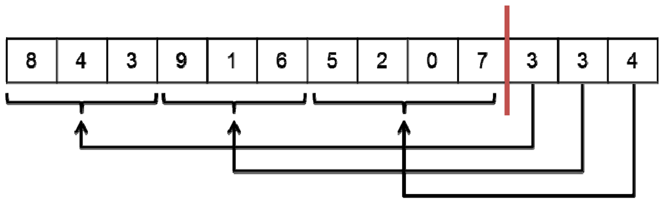

Figure 1. The structure of a sample chromosome in the proposed algorithm

A. Fitness Function

In the proposed algorithm, reducing the total distance traversed and the cost of traversing the path by each of the salesmen have been considered as chromosome fitness. According to the definitions and hypotheses having been explained in the problem statement section and for calculating the fitness of one chromosome, the total distance between cities traversed by each salesman and cost required to traverse those cities by the salesman are calculated through traversal of chromosome from left to right. Finally, the fitness of a salesman, who includes the longest distance and highest cost, is considered as chromosome fitness. The less the chromosome fitness is, the better that chromosome's fitness will be. Equation (1) shows the fitness value of the chromosome in the proposed algorithm.

$$F(x) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{(x_{j+1}-x_j)^2+(y_{j+1}-y_j)^2} + \sum_{i=1}^{m}\sum_{j=1}^{n} \mathrm{cost}(c_j, c_{j+1}) \quad (1)$$

where $\sqrt{(x_{j+1}-x_j)^2-(y_{j+1}-y_j)^2}$ is indicative of the distance between two cities j and j+1 (we suppose have n cities and m salesmen) and $\mathrm{cost}(c_j, c_{j+1})$ will indicate the cost of travel between these two cities. At the end, after calculating the fitness value for each of the salesmen, we will find the salesman with the highest fitness value and will add his fitness value with the total fitness of chromosome. In some algorithms such as [19] only the distance between cities has been used to calculate the chromosome fitness, while the cost of traveling between cities has been also used in the proposed algorithm to be able to model a greater variety of real-world problems. In [19], the fitness levels will obtain from the following Eq. (2):

$$F(x) = \frac{90000}{\sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{(x_{j+1}-x_j)^2+(y_{j+1}-y_j)^2}} \quad (2)$$

B. The Selection Operator

In the proposed algorithm, roulette wheel method is used to select parents that its idea is the probability of selection. The roulette wheel selection method is that at first, the selection probability corresponding with each chromosome is calculated based on its fitness and is obtained using Eq. (3):

$$P_k = \frac{F(k)}{\sum_{i=1}^{n} F(i)} \quad (3)$$

where P_k corresponds probability of selecting chromosome k, f_k denotes the fitness of chromosome k (k-th city). Second, the chromosomes are arranged according to P_k and q_k, which is the cumulative values of P_k, is obtained using the Eq. (4):

$$q_k = \sum_{z=i}^{k} p_z \quad (4)$$

Then, a random number is produced between one and zero for selection of each chromosome, No matter in which range is the mentioned number; its corresponding chromosome will be selected. The roulette wheel selection (or, Fitness proportionate selection) is a method that selects members proportional to their accordance ratio. This method simulates a roulette wheel to determine which members have the chance of reproduction. Proportional to its accordance ratio, each member allocates some parts of roulette wheel to itself. Then at each stage of selection, a member is selected and this process is repeated until enough pairs are selected to form the next generation. Therefore, in this method of selection, all the chromosomes have the chance of being selected.

C. Composition Operator

In the proposed algorithm, two types of exchanges are intended to maintain the scattering of chromosomes.

1) Single-Point Exchange Operator

In this type of exchange, at first a salesman is selected randomly; second, one point between that salesman's cities is selected randomly; then, the salesman's cities from the beginning to the randomly selected number is transformed to the offspring; and finally, the genes of considered salesman are traversed from randomly selected point onwards and transferred to the offspring. Fig. 2 indicates the application of single-point exchange operator.

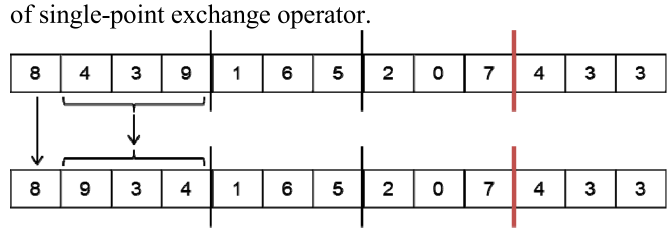

Figure 2. The application of single-point exchange operator.

2) Multipoint Exchange Operator

In this type of exchange, at first, a salesman is selected randomly and then a few points from amongst the cities belonging to the considered salesman are selected randomly and the multipoint integration will be performed. Fig. 3 will show the application of multipoint exchange operator if the tow points have been selected randomly.

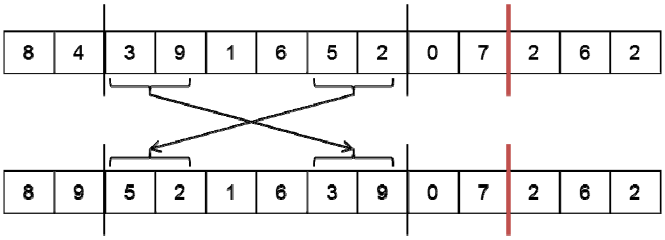

Figure 3. The application of multipoint exchange operator.

D. *The Mutation Operator*

After selecting a parent chromosome for mutation, we will operate in one of the three following methods:

- Displacement method: in this method, two homes of the chromosome in the range of one of the salesmen are selected randomly and their inner values are displaced with each other. The order of one salesman's cities can be changed using this method.
- Transfer method: in this method, two homes of the chromosome belonging to two different salesmen are selected randomly and the numbers of salesmen's two cities are randomly exchanged with each other. Using this method, cities can be displaced between different salesmen.
- The salesman mutation method: in this method, two salesmen are selected randomly and some of the cities belonging to the first salesman are transferred to the second salesman. Using this method, the number of cities belonging to each salesman can be reduced or increased.

The main objective of applying such mutation is to create diversity in the new generation and to exchange the cities between different salesmen. As a result, each chromosome will find a better response in the problem space. The following figures (i.e., Fig. (4)-Fig. (6)) show how mutation operator is applied using three methods – displacement, transfer, and salesman mutation.

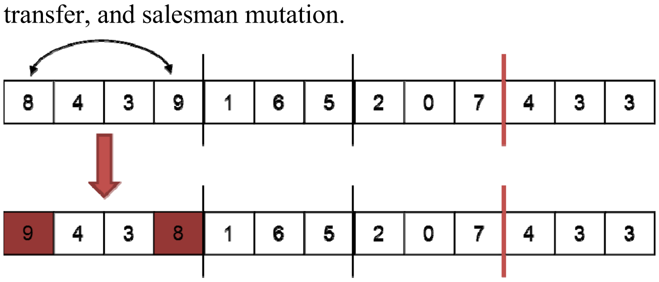

Figure 4. The mutation operator in the proposed algorithm by means of displacement method.

E. *Optimizing Population*

In this stage and prior to the selection of chromosomes for transferring to the next generation, a percentage of current population's chromosomes are optimized using GELS algorithm and then are added to the population as new offspring. In the proposed algorithm, a few chromosomes are randomly selected in an iteration loop and the gravity force algorithm will be recalled for them. In each iteration, gravity force algorithm receives one of the genetic chromosomes as Current Solution (CU) and begins to optimize the chromosome through applying its operators in several stages.

In the proposed gravity algorithm, the process of searching in the problem space is that at first, the neighborhood space of the current solution is created and then a combination of first and second methods, respectively 10% of the first and 90% of the second method are used to calculate the Gravity Force. Furthermore, a combination of multiple and single has been used for transferring in the problem space.

The definition of neighborhood in the GELS algorithm is that we consider each of the salesmen as a cluster and look for the best combination of clusters in which the traversed distance and spend cost is divided equally between all salesmen. Therefore, the proposed algorithm – respectively in GA and GELS algorithm – mostly focuses on finding the best order between each salesman's cities and the fairly division of cities between salesmen. The fitness of solution in the GELS algorithm is obtained using Eq. (5):

$$F(x) = \sum_{i=1}^{m} \sum_{j=2}^{n} \sqrt{(x_1 - x_j)^2 + (y_1 - y_j)^2} + \sum_{i=1}^{m} \sum_{j=2}^{n} \mathrm{cost}(c_1, c_j) \qquad (5)$$

Equation (5) is very similar to Eq. (1) except that the cost and distance between each salesman's first cities with his other cities is calculated instead of calculating the distance and cost of one salesman's cities. At the end, the GELS algorithms turns its response, called Best Solution, to GA. Neighbors are selected based on the defined number for chromosome to be optimized as much as possible. This procedure will be applied on all chromosomes of exciting population.

F. *Chromosome Selection*

For selecting chromosomes for next generation in the proposed algorithm, at first the chromosome has been arranged based on fitness. Then a percentage of chromosomes based on better fitness and other chromosomes have been selected randomly for next generation. In this method of selecting chromosomes for next generation, the scattering of chromosomes in the population has always been maintained and early local optimum will be prevented from occurrence.

G. *Termination Condition*

The termination condition in the proposed algorithm is no improvement of chromosomes in a number of pre-determined generations. That is, if the algorithm is repeated for n specific times from the last time that the algorithm has found a new response but not to be able to find an optimal solution, the algorithm is stopped and the best chromosome will be displayed. Lastly, at first time, initial population is created randomly, based on the genetic algorithm and GELS algorithm. Finally, it evaluated.

TABLE I. COMPARISON RESULTS OF THE GELS-GA, ACO, AND PA IN ORDER TO MINIMIZE THE TOTAL TRAVELLED DISTANCE

Problem				ACO in [22]		SW+As$_{elite}$ in [23]		GELS-GA		%Improvement of the GELS-GA over	
Name	**n**	**m**	**I**	**Avg.**	**Time (s)**	**Avg.**	**Time (s)**	**Avg.**	**Time (s)**	**ACO[21]**	**SW+Aselite[23]**
pr76	76	5	20	180690	51	157562	19	132784	5	26.51	15.73
Pr152	152	5	40	136341	128	128004	41	105205	8	22.84	17.81
Pr226	226	5	50	170877	143	168156	62	152135	9	10.97	9.53
Pr299	299	5	70	83845	288	82195	65	76554	11	8.7	6.86
Pr439	439	5	100	165035	563	162657	95	146523	16	11.22	9.92
Pr1002	1002	5	220	387205	2620	381654	186	354341	27	8.49	7.16
									Minimum	8.49	6.86
									Mean	14.79	11.17
									Maximum	26.51	17.81

TABLE II. COMPARISON RESULTS OF THE GELS-GA AND GA IN ORDER TO MINIMIZE THE TOTAL TRAVELLED DISTANCE AND TIME BY A SALESMAN

Problem		Enhanced GA [24]				GELS-GA				%Improvement of the GELS-GA over
Name	**m**	**Best**	**Avg.**	**Worst**	**Time (s)**	**Best**	**Avg.**	**Worst**	**Time (s)**	**Enhanced GA**
mTSP-51	3	447.42	448.5	449.62	7.10	252	254.5	257	5	43.25
	5	476.11	478.41	482.41	8.78	259	263.5	268	4	44.92
	10	583.57	587.39	589.86	11.20	324	328	332	9	44.15
mTSP-100-II	3	22366.57	22466.41	22611.24	17.27	17800	18035	18270	12	19.72
	5	23895.38	24040.57	24095.96	20.18	20532	20589	20646	14	14.35
	10	27675.42	28033.53	28216.64	26.52	24354	24803.5	25253	17	11.52
	20	39993.83	40274.58	40582.55	34.89	35142	35969	36796	23	10.69
mTSP-150- II	3	39179.41	39361.04	39557.43	28.04	37600	37933.5	38267	25	3.62
	5	40437.18	40663.31	40803.15	34.02	38132	38336.5	38541	28	5.72
	10	44088.29	44546.77	44782.05	44.32	41213	41739	42265	32	6.30
	20	55959.70	56417.86	56572.87	57.13	51393	51443	51493	36	8.81
	30	71605.25	71808.99	71923.98	67.47	66474	66824.5	67175	44	6.94
									Minimum	3.62
									Mean	18.33
									Maximum	44.92

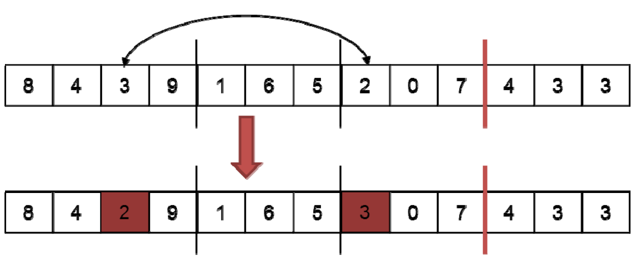

Figure 5. The mutation operator in the proposed algorithm by means of transfer method.

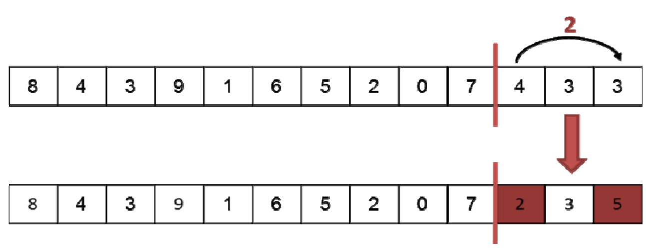

Figure 6. The salesman mutation operator in the proposed algorithm by means of transfer method.

IV. PERFORMANCE EVALUATION

The GELS-GA, was coded in C# language on a Ci3, 2.2 GHz CPU with a 1024 MB RAM. Simulation results are compared to the other existing optimization algorithms in the literature. Standard issues of the mTSP are also included in the GELS-GA simulations, which are derived from the TSPLIB library and articles in [19, 21]. Table 1 compares the GELS-GA with the ACO [22] and SW+ AS$_{elite}$ [23] to solve the mTSP. As it can be seen, the average of the total distance computed by the GELS-GA has 14.5 percent improvement over the ACO and 11.7 percent improvement over the SW+ AS$_{elite}$. Hence, the GELS-GA shows better performance compared to the ACO and SW+ AS$_{elite}$ algorithms. Furthermore, the GELS-GA solves the mTSP faster with the minimum travelled distance. Table 2 demonstrates comparison results of the GELS-GA and the enhanced GA [24, 25] to solve the mTSP. As it is shown, average of the total distance computed by the GELS-GA has 19.9 percent improvement over the enhanced GA. Therefore, the GELS-GA has again significant superiority in achieving optimal and acceptable solutions even in highly complex scenarios. In all the aforementioned tables, *Time* is given in seconds and calculated by a Ci3, 2.2 GHz CPU with a 1024 MB RAM. *Avg.* is the average distance for each problem and *%Improvement* is equal to ((avg. value of a compared algorithm - avg. value of the GELS-GA)/avg. value of a compared algorithm) ×100. In overall, as the GELS-GA finds the shortest path between cities travelled by salesmen and solves large-scale problems with a minimum travelled distance and time, these two parameters are important criteria comparing performance of the GELS-GA with the other algorithms in solving the mTSP. Hence, if the GELS-GA provides a shortest travelled distance and terminates faster, it is superior over the other existing optimization algorithms. The GELS-GA also has some random elements within itself, but it does not merely go forward presumptively. Although it uses local search neighborhood for finding a solution, but it will not move always between them in one form. Although, it has specific behavior of the greedy algorithm, it does not always find the best way to search. Based on the simulation results, improvement of the total travelled distances computed by the GELS-GA to solve the mTSP is 11% at least and it will be increased to 27% even in some cases.

V. Conclusion

In this paper, through combination of GA and GELS algorithms, a proposed algorithm has been presented for solving the mTSP that will be more efficient compared with other methods. The purposes include traversing all exciting cities by salesmen so that all cities to be covered by them and minimizing the total cost of traversing cities. Nowadays, regarding to the enlargement of problems and the importance of speed in reaching the answers, the classical methods are not amenable for solving many of the problems and the random search algorithms are used more than searching all aspects of problem solving space. Therefore, research in this area can be valuable and important. In the above paper, using a combination of innovative genetic and gravity algorithms, a new method was presented for solving the mTSP in order to have more efficiency compared to other methods.

Acknowledgment

This work was supported in the framework of the IT4 Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 by operational programme 'Research and Development for Innovations' funded by the Structural Funds of the European Union and state budget of the Czech Republic, EU.

References

[1] A. Ouaarab, B. Ahiod, and X. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Computing and Applications*, vol. 24, iss. 7-8, pp. 1-11, 2014.

[2] E. Kivelevitch and K.C.M. Kumar, "A Market-based Solution to the Multiple Traveling Salesmen Problem," *Journal of Intelligent & Robotic Systems*, vol. 72, iss. 1, pp. 21-40, 2013.

[3] H. MengShu and L. DaiBo, "A novel method for solving the multiple traveling salesmen problem with multiple depots," *Chinese Science Bulletin*, vol. 57, pp. 1886-1892, 2012.

[4] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, pp. 209-219, 2006.

[5] A.I. Ali and J.L. Kennington, "The asymmetric m-traveling salesmen problem: a duality based branch-and-bound algorithm," *Discrete Applied Mathematics*, vol. 13, pp. 259-276, 1986.

[6] Z. Pooranian, A. Harounabadi, M. Shojafar, and N. Hedayat, "New hybrid algorithm for task scheduling in grid computing to decrease missed task," *World Acad Sci Eng Technol.*, vol. 55, pp. 924–928, 2011.

[7] E. Wacholder, J. Han, and R. C. Mann, "A neural network algorithm for the multiple traveling salesmen problem," *Biology in Cybernetics*, vol. 61, pp. 11-19, 1989.

[8] S. Somhom, A. Modares, and T. Enkawa, "Competition-based neural network for the multiple traveling salesmen problem with minmax objective," *Computers and Operations Research*, vol. 26, pp. 395-407, 1999.

[9] E. Baccarelli, N. Cordeschi, and V. Polli, "Optimal self-adaptive QoS resource management in interference-affected multicast wireless networks, " *IEEE/ACM Tran. on Networking*, vol. 21, no. 6, pp. 1750-1759, 2013.

[10] M. Bellmore and S. Hong, "Transformation of multisalesmen problem to the standard traveling salesman problem," *Journal of Association for Computing Machinery*, vol. 21, pp. 500-504, 1974.

[11] S. Hong and M.W. Padberg, "A note on the symmetric multiple traveling salesman problem with fixed charges," *Operations Research*, vol. 25, pp. 871-874, 1977.

[12] G. Finke, A. Claus, and E. Gunn, "A two-commodity network flow approach to the traveling salesman problem," *Congressus Numerantium*, vol. 41, pp. 167-178, 1984.

[13] S. Bhide, N. John, and MR. Kabuka, "A Boolean Neural Network Approach for the Traveling Salesman Problem," *IEEE Tran. on Computers*, vol. 42, pp. 1271-1278, 1993.

[14] G. Laporte, "A concise guide to the Traveling Salesman Problem," *Journal of the Operational Research Society*, vol. 61, pp. 35–40, 2010.

[15] M. Laguna and F. Glover, "Integrating target analysis and tabu search for improved scheduling systems," *Expert Systems with Applications*, vol. 6, iss. 3, pp. 287-297, 1993.

[16] A. Király and J. Abonyi, "A Novel Approach to Solve Multiple Traveling Salesmen Problem by Genetic Algorithm," *Computational Intelligence in Engineering*, vol. 313, pp. 141-151, 2010.

[17] Y. Bai, W. Zhang, and Zh. Jin, "An new self-organizing maps strategy for solving the traveling salesman problem," *Chaos, Solitons& Fractals*, vol. 28, pp. 1082-1089, 2006.

[18] S. Ghafurian abd N. Javadian, "An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems," *Applied Soft Computing*, vol. 11, iss. 1, pp. 1256-1262, 2011.

[19] S. Sze and W. Tiong, "A Comparison between Heuristic and Meta-Heuristic Methods for Solving the Multiple Traveling Salesman Problem," *International Journal of Computational and Mathematical Sciences*, vol. 1, pp. 200-203, 2007.

[20] G. Reinelt, "TSPLIB- a traveling salesman problem library," *ORSA Journal on Computing*, vol. 4, pp. 134-143, 1996.

[21] http://elib.zib.de/pub/mp-testdata/tsp/

[22] X. Wang, D. Liu, and M. Hou, "A Novel Method for Multiple Depot and Open Paths Multiple Traveling Salesmen Problem," in *International Symposium on Applied Machine Intelligence and Informatics*, 2013.

[23] M. Yousefikhoshbakht and M. Sedighpour, "A Combination of Sweep Algorithm and Elite Ant Colony Optimization For Solving The Multiple Traveling Salesman Problem," in *Proc. the Romanian Academy, Series A,* vol. 13, iss. 4, pp. 295-301, 2012.

[24] F. zhao, J. Dong, S. Li, and X. Yang, "An improved genetic algorithm for the multiple traveling salesman problem, " in *Proc. in Chinese Control and Decision Conference (CCDC)*, 2008.

[25] S. Javanmardi, M. Shojafar, D. Amendola, N. Cordeschi, H. Liu, and A. Abraham, "Hybrid Job Scheduling Algorithm for Cloud Computing Environment, " in *Proc. of IBICA*, vol. 303, pp. 43-52, 2014.