International Journal of Industrial Engineering & Production Research (2018)

March 2018, Volume 29, Number 1
pp. 103- 112

DOI: 10.22068/ijiepr.29.1.103 http://IJIEPR.iust.ac.ir/

Random Gravitational Emulation Search Algorithm (RGES) in Scheduling Traveling Salesperson Problem

Seyed Ahmad Sheibat Alhamdy *& Alireza Hosseinzadeh Kashani

Seyed Ahmad Sheibat Alhamdy, *Associate Professor, Department Of Industrial Management, Firoozkooh Branch, Islamic Azad University*
Alireza Hosseinzadeh Kashani, *PhD Student, Industrial Management, North-Tehran University*

KEYWORDS

Gravitational emulation search**,**
Force of gravity,
Traveling salesperson problem.

ABSTRACT

this article proposes a new algorithm for finding a good approximate set of non-dominated solutions to tackle the generalized traveling salesperson problem. Random gravitational emulation search algorithm (RGES) is presented to solve the traveling salesman problem. The algorithm is based on random search concepts and uses two parameters: speed and force of gravity in physics. The proposed algorithm is compared with genetic algorithm, and experimental results show that the proposed algorithm has better performance and less runtime to be answered.

© 2018 IUST Publication, IJIEPR. Vol. 29, No. 1, All Rights Reserved

1. Introduction

GSA [3] is a novel optimization algorithm that follows the law of gravity and simulates Newton's gravitational force behaviors, introduced by Rashedi [4]. In the algorithm, agents are regarded as objects whose performance is determined using their masses; all these objects attract each other by the gravity force, causing a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication through gravitational force. The heavy masses–which correspond to good solutions – move more slowly than lighter ones. This guarantees the exploitation step of the algorithm. The position of the mass corresponds to a solution to the problem, and its gravitational and inertial masses are determined using a fitness function. Today, some factors can be counted as influential in promoting corporations' efficiency in goods and raw material delivery or transiting that are as follows: (1) the development of manufacturing industry, (2) strong global competition among companies and corporations, (3) the transient life cycle of goods, and (4) the necessary time for marketing and meeting various needs of customers in different places with different distances from manufacturing place. These transiting and delivery systems must be able to deliver goods to the customers on time with the least cost and in the least possible amount of time. Typical applications in operations research include vehicle routing, computer wiring, cutting wallpaper, job sequencing, and job scheduling. Recognizing methods for solving TSP and finding appropriate methods can be one of the important research areas. Due to differences between metaheuristic algorithms, a random gravitational emulation search algorithm as a population-based-algorithm is used to find the shortest path in travelling salesperson problem [23]. One more general form of the famous traveling salesperson problem

* **Corresponding author: *Seyed Ahmad Sheibat Alhamdy***

Email: andishesaaz@gmail.com
Received 5 August 2017; revised 4 December 2017; accepted 26 December 2017

(TSP) is the multiple traveling salesperson problems (mTSP) that consist of finding a collection of travels for m salespersons who start from one specific depot and return to it. In addition, the multi-depot, multiple traveling salesperson problem (MmTSP) is the general form of a single-depot multiple traveling salespersons problem, whereas there should be multiple salespersons in each depot. Since the model cannot be solved within a large-sized instance by operation research software, a metaheuristic algorithm is developed [25]. In fact, once, some special forms of probabilistic constraint have been converted into their equivalent deterministic form, and then a nonlinear constraint has been generated [26]. The problem of this research determines some specific numbers of travels for each travelling salesperson as he is obliged to return to the depot where he first started his travel. This problem is known as a fixed depot traveling salesperson one.

2. Literature Review

Russel [5] introduced the first heuristic methods for solving m travel in TSP was with some limitations. Although the solving method was based on the simple converted TSP and its development on a graph, its algorithm was the expanded model of heuristic method given earlier by Lin and Kernighan [10] for TSP. Potvin et al. [11] introduced another heuristic method based on the exchange procedure for mTSP. Fogel first framed the Parallelism Processing Approach to solving mTSP by the use of evolutionary programming [12]. There are two salespersons with one-purpose traveling. This approach minimizes the variance between the undertaken travels for each salesperson. Some problems containing the limitation of 25 or 50 cities were solved by this improving method, showing some suitable responses approximate to optimum. Wachodler et al. [13] expanded Hopfield-Tank ANN model for mTSP. However, this model is considered complex due to the disability to guarantee feasible solutions [14]. Hsu et al. [15] offered one Neural networks Approach to solving mTSP based on solving m problems of standard TSP. These researchers pointed out that they achieved better results compared with what Wachodler et al. did. One self-organized approach to neural networks for mTSP is related to Wakhotinsky and Golden [16], presented based on Elastic Net approach to TSP problem. Recently, Modares et al. [17] and Somhom et al. [14] introduced one self-organized NN approach to mTSP with the purpose of minimizing travelling time and cost of the most expensive travels among salespersons. Zhang et al. [1] first presented the genetic algorithm (GA) for solving mTSP. One recent use of this is related to Tang et al. [18] who used the genetic algorithm to solve the expanded model of mTSP to program hot rolling Scheduling. The solving process is executed as follows: first, this problem is modelled in the form of mTSP. Then, this problem is converted to a simple TSP. Finally, one refined genetic algorithm is applied to find the respond. Yu et al. [2] also used the genetic algorithm for solving mTSP in designing the travels. Ryan et al. [3] used the taboo search algorithm (TS) for mTSP by an open timing. These researchers presented one integer numerical programming formulation solved by TS algorithm in the format of discrete event simulation (DES). Recently, Song et al. [19] suggested a simulated annealing approach to mTSP with fixed costs for salespersons. This approach was applied to mTSP for 400 cities and 3 salespersons that responded in a suitable timetable. Gomes and Von Zuben [20] offered one method based on the Neuro-Fuzzy system to solve mTSP used in capacitated VRP, which is a network approach based on Fuzzy limitations. Sofge et al. [21] performed and compared some developed algorithms to solve mTSP that consist of neighborhood attractor schema, Shrink-Wrap algorithm for Local neighborhood Optimization, particle Swarm optimization, Monte-Carlo Optimization, Genetic Algorithm, and other developing strategies. Table 1 illustrates various solving methods of mTSP:

Tab. 1. mTSP Solutions

Problem approach	Solving methods
Exact solution methods	Linear programming formulas [4], [5] Cutting planes [6] Branch and bound [8], [28] Lagrange's method with branch and bound [7]

Heuristic methods	Simple Heuristic methods [9], [11] Evolutional algorithms [12] Simulated Annealing (SA) [19] Taboo search (TA) [3] Genetic algorithms (GA) [1], [2], [18] Neural networks (NN) [13], [14], [15], [16], [17]
Converts	Asymmetric mTSP to asymmetric TSP [22] Symmetric mTSP to symmetric TSP [23], [24], [25] MmTSP to TSP [26], [27]

3. MetaHeuristic Algorithms

Heuristic algorithms are stochastic global optimization methods which, in recent years, have been widely used for numerical and combinatorial optimization, classifier systems, and many other engineering problems. Global numerical optimization problems arise in almost every field of science, engineering, and business. Many of these problems cannot be solved analytically due to an increase in search space, with the problem size and dependency of these algorithms on initial solutions, problem dimensions, etc. Therefore, solving these problems using conventional techniques is impractical. Hence, heuristic algorithms have become popular methods to address optimization problems.

Heuristic algorithms must have a good balance between exploration and exploitation (also termed diversification/intensification) to achieve both efficient global and local Searches. In this way, they can efficiently solve optimization problems. Exploration is the ability to investigate the search space for finding new and better solutions, and exploitation is the ability to look for a near-optimal solution. The abilities of exploration and exploitation in every heuristic algorithm are applied with specific operators. Since each operator has its own abilities of exploration and exploitation, the operators should be hybridized artfully together for a good trade-off between exploitation and exploration. Hence, new operators are designed or available operators are redesigned in order to add specific capabilities to heuristic algorithms for solving some problems. Such improvements can be seen in [1–10]. Rogers et al. [1] demonstrated the important role of the crossover operator in a genetic algorithm, which has a significant effect on the search of the problem space. The improved genetic algorithm in [2] is based on a family tree that evaluates the degree of similarity between chromosomes and validly avoids genetic operations among similar chromosomes when the population size is small. This policy is done to prevent premature convergence. The same-site-copy-first principle as a crossover is introduced for GA in [3]. The new genetic algorithm in [4] became faster with higher precision, using ‘‘multiple’ mutation, localized mutation, and a double crossover. There are several mutation operations for permutation problems such as adjacent two-change, arbitrary two-change, arbitrary three-change and shift change [5]. For permutation-based representations, partially matched crossover (PMX) and linear order crossover (LOX) have been widely used. PMX attempts to keep the absolute positions of elements and LOX with respect to relative positions [6]. Three regulations (immigration, local optimization, and global optimization) were established in [7] based on several empirical optimization strategies to enhance the local optimization capability of GA and its

$$\mathrm{qi}(\mathrm{t}) = \frac{\mathrm{fit}_{\mathrm{i}}\mathrm{t} - \mathrm{worst}(\mathrm{t})}{\mathrm{best}(\mathrm{t}) - \mathrm{worst}(\mathrm{t})} \quad (1)$$

$$\mathrm{M}_{\mathrm{i}}\mathrm{t} = \frac{\mathrm{qi}(\mathrm{t})}{\sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{qj}(\mathrm{t})} \quad (2)$$

convergence speed. The improved algorithm in [8] employs the generation alternation GSA as one of the newest heuristic algorithms introduced by Rashedi et al. [17, 18], which is inspired by the law of gravity and mass interactions along with the implementation of Newtonian gravity and the laws of motion. The fruitful ability of GSA to find and converge to an optimality derives from the results of experiments previously undertaken [17–19]. In this algorithm, the gravitational force guides the masses. As this force absorbs the masses into each other, in case premature convergence happens, there will not be any recovery for the algorithm. In other words, after becoming converged, the algorithm loses its ability to explore and, then, becomes inactive. Therefore, new operators should be added to

GSA in order to increase its flexibility to solve problems that are more complicated. In this case, a new operator, based on astrophysics, is proposed in this paper. This paper is organized as follows. The section ''Gravitational Search Algorithm'' provides a brief review of GSA. In the section ''Disruption Phenomenon in Nature'', the disruption phenomenon is explained. The proposed disruption operator and its characteristics are described in the section ''Simulation of Disruption Phenomenon as a Gravitational Operator''. A comparative study is presented in ''Experimental Results''; finally, in the last section, the paper is concluded.

4. Gravitational Search Algorithm

Rashedi et al. first introduced the Gravitational Search Algorithm (GSA) as a new stochastic population-based heuristic optimization tool [17–19]. This approach provides an iterative method that simulates mass interactions and moves through a multi-dimensional search space under the influence of gravitation. This heuristic algorithm has been inspired by the Newtonian laws of gravity and motion [19]. The effectiveness of GSA and its version for binary encoded problems (BGSA) [20] in solving a set of nonlinear benchmark functions has been proven [19, 20]. Based on [19], in GSA, the mass of each agent is calculated after computing the current population fitness as follows (for a minimization problem):

where N, Mi (t), and fiti (t) represent the population size, the mass, and the fitness value of agent i at t, respectively; worst(t) and best(t) are defined as follows (for a minimization problem):

$$\mathrm{best}(t) = \min\ \mathrm{fit}_j(t) \quad j \in \{1, \dots, N\} \tag{3}$$

$$\mathrm{worst}(t) = \max\ \mathrm{fit}_j(t) \quad j \in \{1, \dots, n\} \tag{4}$$

To compute the acceleration of an agent, total forces from a set of heavier masses applied should be considered based on a combination of the law of gravity and the second law of Newton on motion (Eq. (5)) [19,20]. Afterwards, the next velocity of an agent is calculated as a fraction of its current velocity added to its acceleration (Eq. (6)). Then, its position could be calculated using Eq. (7).

$$a_i^d = \sum_{j \in kbest\ j \neq 1} \mathrm{rand}_j G(t) \frac{M_j(t)}{R_{i.j}(t) + \epsilon} \left(X_j^d(t) - X_i^d(t) \right)$$

$$d = 1,2,\dots,N \quad i = 1,2,\dots,N \tag{5}$$

$$v_i^d(t+1) = \mathrm{rand}_i \times v_i^d(t) + a_i^d \tag{6}$$

$$a_i^d(t+1) = a_i^d(t) + v_i^d(t+1) \tag{7}$$

where a_i^d , v_i^d, and x_i^d present the acceleration, velocity, and position of agent i in dimension d, respectively. $rand_i$ and $rand_j$ are two uniform random numbers at the interval [0, 1], ε is a small value, n is the dimension of the search space, and Ri,j (t) is the Euclidean distance between two agents i and j. kbest is the set of the first K agents with the best fitness value and biggest mass, which is a function of time, initialized to K0 at the beginning and decreased with time. Herein, K0 is set to N (total number of agents) and decreases linearly to 1. G is a decreasing function of time, which is set to G0 at the beginning and decreases exponentially towards zero with lapse of time. It is noted that Xi = (x_i^1 , x_i^2 , . . . , x_i^n) indicates the position of agent i in the search space, which is a candidate solution. Figure 1 gives the pseudocode of the Standard GSA (SGSA).

5. Problem Definition

Some parameters are used in solving the problem defined as follows:

3-1. Decision variable:

This model is based on the 3-indicator x_{ijk} decision variable as follows:

$$x_{ijk} = \begin{cases} 1 & \text{If a salesperson who exits from depot k crosses arc (i,j)} \\ 0 & \text{Otherwise} \end{cases}$$

This variable considers 1 for those crossed arc in the travels, and 0 for other arcs. In fact, considering this binary variable, those arcs from which no salesperson has ever crossed are deleted for the response. The result can be obtained by concluding other variable arcs in the response. Since each salesperson should return to the depot where he exists, the third indicator k is considered for depicting the dependence if each arc returns to one depot in this decision variable.

3-2- Input parameters:

c_{ij}: Distance between cities i and j

N: the number of total locations

M: The number of salespersons

d: the number of depots

M_k: the number of salespersons in depot k

V: The total locations (cities)

V': The total intermediate locations (intermediate cities)

D: Total depots

L: the maximum number of intermediate cities for each salesperson to visit
K: the minimum number of intermediate cities for each salesperson to visit
u_i: The total visited cities for each salesperson before arriving at city i (including depot)

6. Statement of The Problem

Consider the complete directed graph $G=(V,A)$ in which V is the total n points (vertices), A is the total arcs, and $C=(c_{ij})$ is the cost matrix (distances) for each arc $(i,j)\in A$. The cost matrix C could be symmetrical, asymmetrical or Euclidean. The total set of points is divided as in V=V'UD and d is the first point from V which makes the set of depots (D). At first, there are m_k salespersons in depot k, while the total set of salespersons is m. The total intermediate points (cities) include:
V'= {d+1, d+2… n}
We have a binary variable with 3 indicators x_{ijk} (notice that x_{iik} equals zero):

$$x_{ijk}=\begin{cases}1 & \text{If a salesman who exits from k depot crosses (i,j) arc}\\ 0 & \text{Otherwise}\end{cases}$$

u_i indicates the total points which exist along the travel from depot to point i for each salesperson (concluding depot), and L is the highest number of points where a salesman must visit. Thus, for each $i\geq 2$, we have $1\leq u_i\leq L$.
k is also the least numbers of points where one salesperson must visit. It means that if x_{ijk} = 1, $K\leq u_i\leq L$ must be sustained. Figure 1 shows one mTSP problem with 5 cities and 2 salespersons. In addition, Figure 2 illustrates one mTSP problem with 10 cities and 4 salespersons, 2 depots, and 2 salespersons in each depot, indicating one response with 4 travels (one travel for each salesperson):

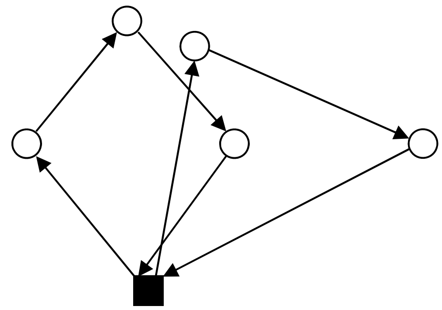

Fig. 1. mTSP with 5 cities and 2 salespersons

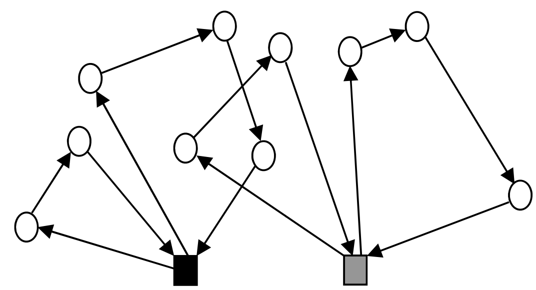

Fig. 2. mTSP with 10 cities and 4 salespersons , 2 depots, and 2 salespersons in each depot

7. Formulation

These formulations for some traveling salespersons with fixed depots and destinations are as follows:

$$\textit{Minimize}\quad \sum_{k\in D}\sum_{j\in V'}(c_{kj}x_{kjk}+c_{jk}x_{jkk})+\sum_{k\in D}\sum_{i\in V'}\sum_{j\in V'}c_{ij}x_{ijk}$$

S.t.

$$\sum_{j\in V'}x_{kjk}=m_k,\qquad k\in D, \tag{1}$$

$$\sum x_{kjk}+\sum\sum x_{ijk}=1,\qquad j\in V', \tag{2}$$

$$x_{kjk}+\sum_{i\in V'}x_{ijk}-x_{jkk}-\sum_{i\in V'}x_{jik}=0,\qquad k\in D,\qquad j\in V', \tag{3}$$

$$\sum_{j\in V'}x_{kjk}-\sum_{j\in V'}x_{jkk}=0,\qquad k\in D, \tag{4}$$

$$u_i+(L-2)\sum_{k\in D}x_{kik}-\sum_{k\in D}x_{ikk}\leq L-1,\qquad i\in V' \tag{5}$$

$$u_i + \sum_{k \in D} x_{kik} + (2-K)\sum_{k \in D} x_{ikk} \geq 2, \quad i \in V' \tag{6}$$

$$\sum_{k \in D} x_{kik} + \sum_{k \in D} x_{ikk} \leq 1, \quad i \in V', \tag{7}$$

$$u_i - u_j + L\sum_{k \in D} x_{ijk} + (L-2)\sum_{k \in D} x_{jik} \leq L-1, \quad i \neq j, \quad i, j \in V', \tag{8}$$

$$x_{ijk} \in \{0,1\}, \quad i, j \in V' \quad k \in D. \tag{9}$$

$$m \geq 1 \text{ and integer} \tag{10}$$

This formulation is valid provided that $2 \leq K \leq \lfloor n/m \rfloor$ and $K \leq L \leq n-(m-1)K$ can be feasible. Thus, one salesperson cannot leave one depot, visit only one city, and return to the same depot. As a result, the number of cities one salesperson must visit, without considering the depot, must be at least 2 cities. The first limitation guarantees that m_k salespersons exactly exist in each $K \in D$ depot. The second limitation indicates that each point (city) is exactly visited once. The unity of travels for intermediate points and depots is illustrated by the third and fourth limitations, respectively. Limitations (5) and (6) apply the upper and lower limits of visited points to the travels. In general, if i is the first point in each travel, these limitations set u_i equal to 1. The travels with only one intermediate point are prohibited by limitation (7). Finally, limitation (8) is sub tour elimination constraint (SEC) that avoids the existence of any sub-travels (under-travel) among intermediated points. These sub-travels include some blocked travels, which are made without any depot as start or end point by intermediate points. These could appear if there would be no suitable SEC in responses. It obviously appears that this problem is an NP-Hard one. In this formulation, elimination is made in terms of the number of $O(dn^2)$ binary variable and $O(n^2)$.

The proposed algorithm: The proposed algorithm mimics the random gravitational force (RGES) as a strategy to solve TSP problem. The purpose of this algorithm to find the shortest path between cities by Sellers and ability to solve the large-scale problem in minimum time. Figure 2 shows the proposed algorithm.

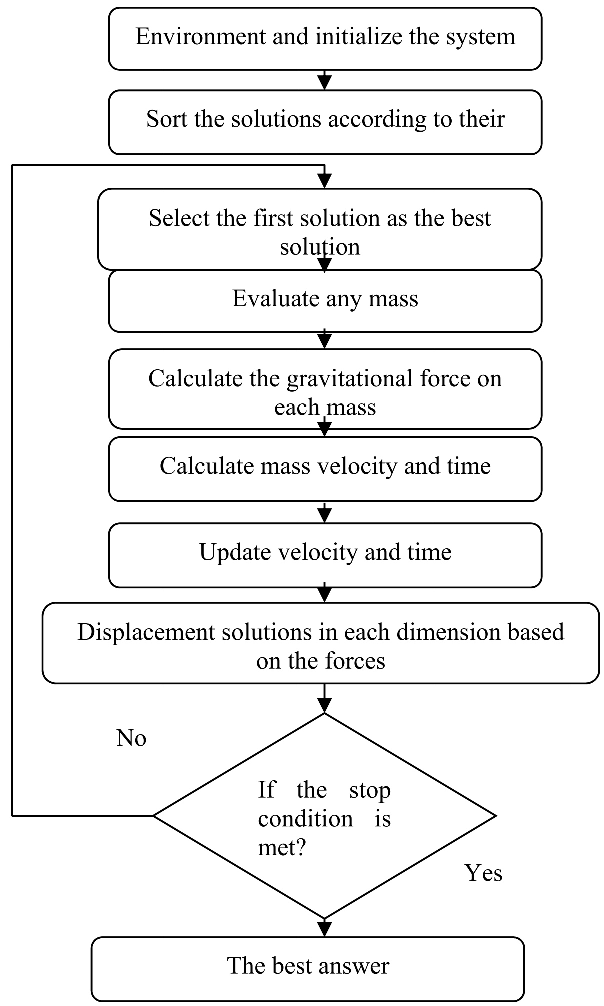

Fig. 2. structure of the proposed algorithm

Solution dimensions: In the proposed method, each dimension can be considered equivalent to a city, and the number of response dimensions is equal to that of input cities. Neighboring of the current response represents a city with the minimum distance and maximum speed along with response city from which no clerk has crossed from it yet.

Neighboring: Unlike other search algorithms in RGES algorithm, neighbors are not a random search; however, each current solution has several neighbors where each of them, according to a specific change of direction, comes from the neighbor mentioned. The proposed city with minimum distance and maximum speed of response is now considered as neighbors, and candidates are selected.

Experimental result: To implement the algorithms, the Matlab programming language was used. Applications were carried out through a computer with 1 GB of RAM and CPU 2.4 GHz Pentium-IV. The proposed algorithm (TSP_RGES) was compared with the genetic algorithm (GA). The comparison of algorithms was made for the five sets of test data to the systems designed for small, medium, and large covers. The test data are named as Test_sh_c. Sh is the issue number and c is the number of cities.

Table 1 compares the results of both algorithms GA and TSP-RGES. This Table shows that implementation time for the proposed algorithm is less than that for the GA algorithm. The results show that the proposed algorithm needs much less time than the GA algorithm does, for solving.

Tab. 1. The comparison results of GA and TSP-RGES algorithms

Problem dimension	TSP-RGES			GA		
	Time	Fitness best	Fitness worst	Time	Fitness best	Fitness worst
Test_01_20	very Negligible	130	234	20.966	366	540
Test_02_50	Negligible	296	497	39.966	437	642
Test_03_100	0.014	580	1086	204.286	972	1483
Test_04_500	0.566	2502	3705	1173.831	3040	5126
Test_05_1000	4.353	5659	8422	2256.090	6853	9734

Fitness best: Best value for the algorithm returns the number of cities and the distance between cities.
Fitness worst: worst value for the algorithm returns the number of cities and the distance between cities.
Time: implementing Time of algorithm

Figure 3. Comparison of two algorithms using the test dataset shows a Test_01_100. As seen, very little time is required to implement the proposed algorithm and reach better results.

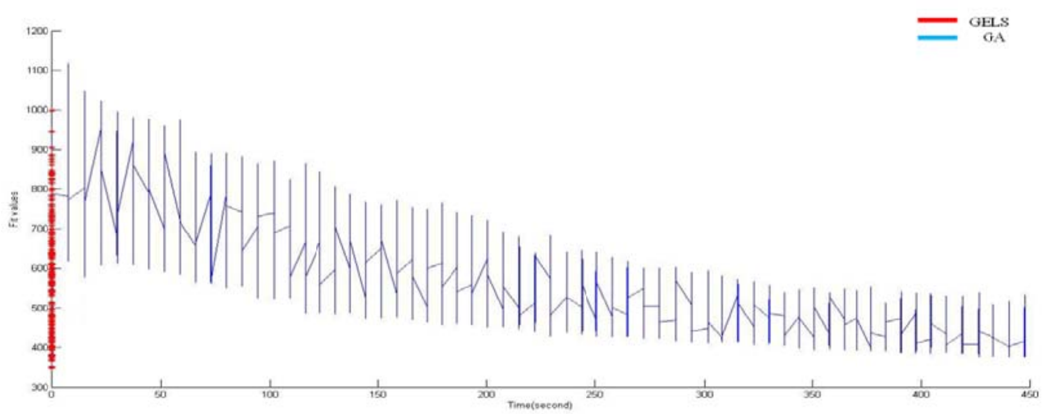

Fig. 3. Comparison of GA and TSP-RGES algorithms (test data test-01-100)

Figure 4 shows the time required for the implementation of two algorithms for test data. This Figure shows that the algorithm requires much less time than the GA algorithm would, for solution.

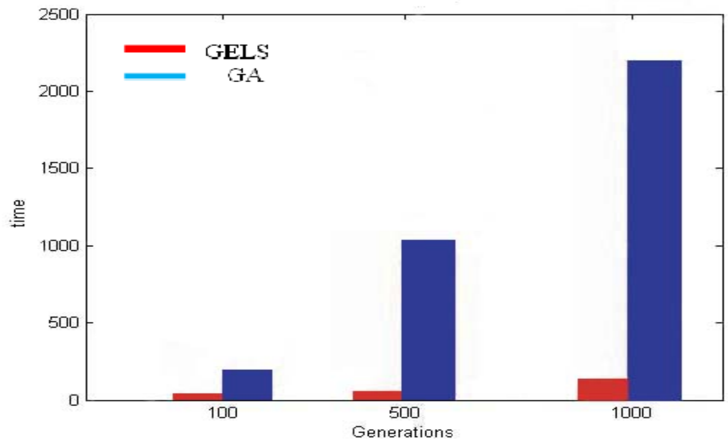

Fig. 4. comparison GA and TSP-RGES algorithm based on runtime

8. Conclusion

In this paper, an algorithm was introduced based on the gravitational force concept for solving the traveling salesperson problem. The purpose of this algorithm was to reduce the runtime and find the shortest route between the cities by the seller. Performance of this algorithm is compared with that of GA algorithm. The results showed that TSP-RGES algorithm, as compared with the GA algorithm, requires much less time to reach better results. This improvement is better reflected in the large-scale systems. For future works, these two algorithms can be combined to develop a hybrid algorithm. Due to the importance of the time required to reach the final answer and of finding the shortest path in the traveling salesperson problem, less runtime in the proposed algorithm is one of its advantages. Minárová study (2018) replaced the product of the masses in the Gravitational search algorithm introduced by other bivariate functions with specific properties. Then, the properties of these functions are analyzed to guarantee convergence in the algorithm, and an application to justify our theoretical study and the need of using functions other than the product are discussed [24].

References

[1] Hongwei Moand and Lifang Xu,"Biogeography Migration Algorithm for Traveling Salesman Problem," ICSI, (2009), pp. 405-414.

[2] Foziahanif khan, Nasiruddin khan and syedinayatullah, “Solving TSP problem BY using genetic algorithm.”, International Journal of Basic & Applied Sciences IJBAS, Vol: 9. No. 10, (2009).

[3] Nagham Azmi AL-Madi and Ahamad Tajudin Khader,"The Traveling Salesman Problem as a Benchmark Test for a Social-Based Genetic Algorithm," Journal of Computer Science, Vol. 4, No. 10, (2008), pp. 871-876.

[4] Hejazi, Seyyed, Reza and Soltani, Reza, “The implementation of combined ant colony and genetic algorithm for solving traveling salesman problem”, 4th International Industrial Engineering Conference, (2005).

[5] Alessio Plebe and Angelo Marcello Anile,"A NeuralNetwork Based Approach to the Double Traveling Salesman Problem", This is a preprint of the extended article on Neural Computation, (2001).

[6] Zhenchao Wang, HaibinDuan, and Xiangyin Zhang, "An Improved Greedy Genetic Algorithm for Solving Travelling Salesman Problem", Fifth International Conference on Natural Computation, (2009).

[7] Voudouris, chris, Edward Tsang, "Guided Local Search", Technical Report CSM-247, Department of Computer Science, University of Essex, UK, August (1995).

[8] Barry Lynn Webster, "Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational

Attraction", Ph.D., thesis, Melbourne, Florida Institute of Technology, May (2004).

[9] S. Raja Balachandar and K. Kannan," Randomized gravitational emulation search algorithm for symmetric traveling salesman problem", Department of Mathematics, SASTRA University, Thanjavur 613 402, India, (2007).

[10] Esmat Rashedi, Hossein Nezamabadi-pour and Saeid Saryazdi,"GSA: A Gravitational Search Algorithm Department of Electrical Engineering", Shahid Bahonar University of Kerman, P.O. Box 76169-133, Kerman, Iran (2009).

[11] Jiang, Y., Hu, T., Huang, C. and Wu, X. ''An improved particle swarm optimization algorithm'', Applied Mathematics and Computation, Vol. 193, No. 1, (2007), pp 231-239.

[12] Fourie, P.C. and Groenwold, A.A. ''The particle swarm optimization algorithm in size and shape optimization'', Structural and Multidisciplinary Optimization, Vol. 23, No. 4, (2002), pp. 259-267.

[13] Baskar, G. and Mohan, M.R. ''Contingency constrained economic load dispatch using improved particle swarm optimization for security enhancement'', Electric Power Systems Research, Vol. 79, No. 4, (2009), pp. 615-621.

[14] Yuan, X., Nie, H., Su, A., Wang, L. and Yuan, Y. ''an improved binary particle swarm optimization for unit commitment problem'', Expert Systems with Applications, Vol. 36, (2009), pp. 8049-8055.

[15] Chang, W.D. and Shih, S.P. ''PID controller design of nonlinear systems using an improved particle swarm optimization approach'', Communication in Nonlinear Science and Numerical Simulation, Vol. 15, No. 11, (2010), pp. 3632-3639.

[16] Arumugam, M.S. and Rao, M.V.C. ''On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing Optimal control of a class of hybrid systems'', Applied Soft Computing, Vol. 8, No. 1, (2008), pp. 324-336.

[17] Rashedi, E. ''Gravitational search algorithm'', M.Sc. Thesis, Electrical Engineering Department, Shahid Bahonar University of Kerman, Iran (2007) (in Farsi).

[18] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S. and Farsangi, M.M. ''Allocation of static var compensator using gravitational search algorithm'', Proceedings of the First Joint Conference on Fuzzy and Intelligent Systems, Mashhad, Iran (Aug. 2007).

[19] Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S. ''GSA: a gravitational search algorithm'', Information Science, Vol. 179, No. 13, (2009), pp. 2232-2248.

[20] Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S. ''BGSA: binary gravitational search algorithm'', Natural Computing, Vol. 9, No. 3, (2010), pp. 727-745.

[19] Harwit, M. The Astrophysical Concepts, 3rd ed., NewYork (1998).

[20] Yao, X., Liu, Y. and Lin, G. ''Evolutionary programming made faster'', IEEE Transaction on Evolutionary Computation, Vol. 3, No. 2, (1999), pp. 82-102.

[21] Haupt, R.L. and Haupt, E., Practical Genetic Algorithms, 2nd ed., John Wiley & Sons (2004).

[22] Bergh, F.V.D. and Engelbrecht, A.P. ''A study of particle swarm optimization particle trajectories'', Information Sciences, Vol. 176, No. 8, (2006), pp. 937–971.

[23] Yaghini, M. Momeni, M. Sarmadi,m,2011,Finding the Shortest Hamiltonian Path for Iranian Cities Using Hybrid Simulated Annealing and Ant Colony Optimization Algorithms, International Journal off Industtrial Engiinering & Production Research, Vol. 22, No. 1, (2011), pp. 31-42.

[24] M. Minárová, D. Paternain, A. JuriobJ. Ruiz-Aranguren, Z. TakáčaH. Bustince, 2018, Modifying the gravitational search

algorithm: A functional study, Information Sciences Vol. 430–431, (2018), Pages 87-103.

[25] adeleh behzad.mohammad ali pirayesh,mohammad ranjbar,routing and production scheduling for a mobile factory, International Journal off Industrial Engineering & Production Research, Vol. 28, No. 3, (2017), pp. 299-308.

[26] M. Heydari, B. Karimi & S.A. Torabi,2009,The Economic Lot Scheduling Problem in Flow Lines with Sequence-Dependent Setups, International Journal of Industrial Engineering & Production Research, Vol. 20, Nos. 1 & 2, (2009).

Follow This Article at The Following Site

Sheibat Alhamdi A, Hosseinzadeh Kashani A. Random gravitational emulation search algorithm (RGES (in scheduling traveling salesman problem. IJIEPR. 2018; 29 (1) :103-112
URL: http://ijiepr.iust.ac.ir/article-1-795-en.html