Turkish Journal of Electrical Engineering and Computer Sciences

Volume 28 | Number 5 Article 35

1-1-2020

An improved memetic genetic algorithm based on a complex network as asolution to the traveling salesman problem

HADI MOHAMMADI

KAMAL MIRZAIE

Mohammad Reza Mollakhalili Meybodi

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 2910 – 2925
© TÜBİTAK
doi:10.3906/elk-1911-106

An improved memetic genetic algorithm based on a complex network as a solution to the traveling salesman problem

Hadi MOHAMMADI, Kamal MIRZAIE*, Mohammad Reza MOLLAKHALILI MEYBODI
Department of Computer Engineering, Maybod Branch, Islamic Azad University, Maybod, Iran

Received: 16.11.2019 • **Accepted/Published Online:** 29.05.2020 • **Final Version:** 25.09.2020

Abstract: A genetic algorithm (GA) is not a good option for finding solutions around in neighborhoods. The current study applies a memetic algorithm (MA) with a proposed local search to the mutation operator of a genetic algorithm in order to solve the traveling salesman problem (TSP). The proposed memetic algorithm uses swap, reversion and insertion operations to make changes in the solution. In the basic GA, unlike in the real world, the relationship between generations has not been considered. This gap is resolved using the proposed complex network to allow selection among possible solutions. The degree measure has been used for analysis the network. Different scenarios have been evaluated to solve seven TSPLib problems. For example, the results indicated that the memetic algorithm with a complex network, the memetic algorithm with the proposed local search and basic GA have 0.31%, 1.15% and 38% errors, respectively, when solving the TSP for 70 cities compared to the best solution in the TSPLib database. These results offered better performance of the memetic algorithm with a complex network compared to the memetic algorithm with the proposed local search and the basic GA. Also, the average run time of the algorithms showed their scalability.

Key words: Travelling salesman problem, genetic algorithm, memetic algorithm, local search, complex network

1. Introduction

The current study solved the traveling salesman optimization problem using the proposed algorithms. This is a basic problem in routing and transportation planning. The goal of this problem is to determine the shortest route among a set of cities in such a way that each city is visited exactly once and the route returns to the city of origin[1]. The traveling salesman problem is a Hamiltonian cycle in which each route represents a permutation from the city of origin to all other cities. If n is the number of cities and p is a permutation from the Hamiltonian path, the purpose of solving the TSP is to minimize Formula 1:

$$\sum_{i=1}^{n} d_{ip(i)}, \tag{1}$$

where p(i) is the node number after i in permutation p and d_{ij} is the distance between cities i and j. Recently, both heuristic and metaheuristic algorithms have been used to solve hard problems such as optimization. Unlike more precise methods, these algorithms use a polynomial time order to return a near-optimal solution. The genetic algorithm, ant colony algorithm (ACO), artificial bee colony algorithm (ABC), biogeography-based optimization (BBO), simulated annealing (SA), tabu search (TS) and particle swarm optimization(PSO) are

*Correspondence: k.mirzaie@maybodiau.ac.ir

examples of heuristic and metaheuristic algorithms [2–8]. These algorithms can change the process of algorithm implementation and solution selection using different parameters or operators and have been used to solve optimization problems such as the TSP. The common point in these algorithms is that they change the problem solution using operators, parameters or coefficients and generate different solutions [9–12].

Single-solution algorithms are used alongside algorithms such as the GA and PSO because the original algorithm seeks global optimization and the local search algorithm seeks a local optimum. The combination of these two concepts can generate memetic algorithms [13]. In these algorithms the global and local search, the concepts of exploration and exploitation are addressed. The goal of exploration is to ensure that a search is global and the goal of exploitation is to find best solution around a single solution. This is a kind of local search in which the goal of the memetic algorithm is to create a balance between the two concepts [14]. The basic GA has good exploration abilities because its basis is global when finding solutions. In metaheuristic algorithms, especially population-based algorithms, the level of access of the solutions is individualized. The selected solutions do not have direct relationships and do not affect one another. For example, using the crossover operator and making changes in the parents can generate a new child in the GA. In a basic GA, the possibility of a relationship between the parents and children in the population has not been addressed. In recent years, only a few articles have discussed the gap in the relationship between the solutions [15, 16]. However, complex networks can be used with heuristic and metaheuristic algorithms to address this problem.

2. Background

A GA consists of initial population generation, parent selection, a crossover operator and a mutation operator. Because a GA is a population-based algorithm, it begins by generating an initial population. Darwin's theory is described as survival of the fittest. In this theory, individuals who are the most fit will generate children who are the most fit. In the stage of parent selection, one of the following two methods can be used selection.

Random method: In this method, the parents are randomly selected from the initial population. The advantage of this method is the simplicity of implementation and also the selection of the elite parent based on chance.

Roulette wheel method: In this method, based on the solution rate, a chromosome is added to the solutions of other chromosomes and the distance from zero to the sum of all solutions is generated. This distance is considered to be circular because of its simplicity. Along this distance, a number is randomly selected. The range of the selected number is related to the chromosome which is selected as the elite chromosome. For each chromosome, the selection probability is the calculation of Formula 2:

$$ProCh_i = \frac{fitnessch_i}{\sum_{k=1}^{n} fitnessch_k}. \tag{2}$$

Then, the child is generated in this stage using the crossover operator. The mutation operator is used to improve optimality. Changes are randomly provided in one or more genes of a chromosome sequence. The main methods of implementing the mutation are listed below. Swap operation: Two genes are randomly selected from the chromosome and are swapped in order to generate a new chromosome. Insertion operation: Two genes are randomly selected and the first gene is inserted behind the second gene. In a natural manner, the genes between two selected genes are shifted one step to the left. Reversion operation: Two genes are randomly selected, and these genes and the genes between them are swapped in the reversion form. After the mutation operator, if the total number of iterations are not completed, the algorithm will continue from the selection stage, otherwise the algorithm will end.

3. Complex networks

Complex networks have a set of vertices, edges and important topological properties which are neither completely random nor perfectly regular. Most social, biological, and technological networks can be represented as a complex network. The internet is a network of routers and the world wide web is a network of websites. The brain is a network of neurons. The focus on complex networks has moved from the analysis of small networks to systems with thousands or millions of nodes. The approaches were initiated by Watts and Strogoutz on small-world networks [17] and then by Barabassi and Albert [18] on scale-free networks. The complex networks can be modeled using methods such as regular, random, small-world and scale-free networks. Network theory analysis is used fields such as biology, organizational behavior, social networks and smart data analysis. Different measures are used for network measurement. In these types of networks, information exchange between nodes can improve understanding of complex networks. This includes the average degree of nodes, clustering coefficient, degree distribution function and giant component formation in the network.

4. Related work

Metaheuristic algorithms can be divided into two classes: those without a complex network and those with a complex network. In metaheuristic algorithms without a complex network, the relationship between the solutions are not considered. Most algorithms do not consider this relationship when solving the TSP. The optimization problem is solved either in the form of a heuristic or metaheuristic algorithm or a combination of two or more algorithms. In the current study, a memetic algorithm with a proposed local search has been used to prevent becoming stuck in the local optima. Metaheuristic algorithms based on a complex network form the second group in which the relationship between the solutions generated by different generations is considered when generating a new solution. Only a few papers have been presented in this area. In the following, some methods are presented for these two classes along with brief explanations of their innovations and the algorithm utilized. In [19] used BBO to solve the TSP problem and has performed better in many cases compared to other population-based algorithms. The biogeography optimization algorithm is an evolutionary algorithm presented by Simon [20]. This algorithm is based on the migration of animals and birds between islands. In [21] improved the migration operator in the BBO algorithm to help solve the TSP. In [22] proposed an adaptive PSO algorithm based on a directed weighted complex network called directed weighted complex network particle swarm intelligence (DWCNPSO). The particles are uniformly scattered over the search space using the small-world network topology in order to initialize the position of the particles. At the same time, an evaluation mechanism of the direct dynamic network is used to build the particles on a scale-free network according to the power-law distribution. The proposed method improves algorithm diversity and prevents the particles from falling into the local search. The simulation results indicated that the proposed algorithm prevents early convergence and has a faster convergence rate than other algorithms. In [23] implemented the PSO by interacting with scale-free networks. The scale-free network has been used to represent individual interactions in the population as a class called scale-free particle swarm intelligence (SF-PSO). Unlike a traditional PSO with fully connected or regular topology, SF-PSO uses scale-free topology to combine individual diversity in the search and information dissemination to provide a completely different optimization process. The systematic results for several standard benchmark functions demonstrated that SF-PSO can improve equilibrium between the convergence speed and optimization quality. Analysis has shown better performance for this algorithm than for the traditional PSO algorithm. In [24] made changes in both the crossover and mutation operators of the GA to improve the efficiency and performance of the results when solving the TSP. The results show improved performance of automatic selection of the different crossover and mutation operators compared to the standard operators of the GA. In [25] a new mechanism of the crossover operator in the GA for solving the TSP is presented. They introduced two new crossover operators, modified sequential constructive crossover radius (MSCX_Radius) and random crossover (RX). If the next legitimate node is not found for each parent, the MSCX_Radius operator is the changed form of the MSCX operator in which, at the crossover implementation stage, a node that is more cost-effective from among all nonvisited nodes in both parents is selected. The legitimate node is also the first node after the current node which has not been visited in any parent. The MSCX operator can also change the chromosome. First, two parents with for generating two children are selected. The first node of the first parent is inserted as the first node of the first child to generate the first child. In order to select the next node, the first legitimate node after the current node is chosen in both parents and, among these nodes, the one that is more cost-effective is selected as the next node in the first child. This process continues until the first child nodes are completed [26]. The RX crossover operator randomly inserts half of the genes from the first parent into the first child's chromosomes to generate the first child. The remaining genes are inserted from the second parent into the first child sequence in the same visit order. In order to generate the second child, the same operations are repeated in reverse. The TSP-Lib database was used to compare the results. The results showed improved performance of the proposed algorithm compared to the GA in the minimum, mean and maximum cost states. Qingzheng Xu et al. [27] solved the TSP by applying changes in the solution to the biogeography optimization algorithm. Their proposed algorithm, called oppositional biogeography-based optimization using the current optimum (COOBBO), was presented by Qingzheng xu et al. [28]. Eight well-known TSPs were used to compare the results. The COOBBO algorithm showed improved performance. This is an improved form of the oppositional biogeography-based optimization (OBBO) method in which the current optimum is used at each stage instead of considering the global optimum as the basis of opposition-based learning (OBL) as proposed by Tizhoosh [29]. The idea is that there is an opposite form of any solution, objective function or weight in the optimization problems. By determining this opposite form, it is possible to generate an initial population with higher diversity because of the solution diversity. OBBO has been used to improve the biogeography problem and OBL has been used for the diversity in the population. In [30] used complex networks as self-organization components that can be modeled using methods such as regular, random, small-world and scale-free networks. In [31] presented a methodology for PSO algorithm conversion inside a complex network. Statistical features in the complex networks can be used in the discussed adaptive viewpoint.

5. Proposed algorithms

In the current study, the relationships of the population have been used for building a network. The following algorithms have been used with or without the complex network to solve the TSP.

5.1. Basic genetic algorithm

In the basic genetic algorithm, the normal routine of the GA is followed and the initial population selection, crossover operator, mutation operator and investigation of the stop condition stages are carried out. A single point crossover operator has been used. In the mutation stage, a swap operation is performed that changes only the solution. The gene selected from the chromosome is swapped for another gene and a new solution is generated. One problem with the use of the basic GA is that it slows down as it moves towards the global optimum. The genetic memetic algorithm can be used to solve some of the problems of the GA.

5.2. Memetic genetic algorithm

Combined algorithms, such as the memetic algorithm, are able to perform a local search with knowledge. With the use of a "meme" in a local search in order to represent the current responses at each generation, the memetic algorithm tries to locally improve these responses by providing a balance between exploration and exploitation in the search. These operations provide better performance in the search process. Figure 1 is a flowchart of the proposed memetic GA.

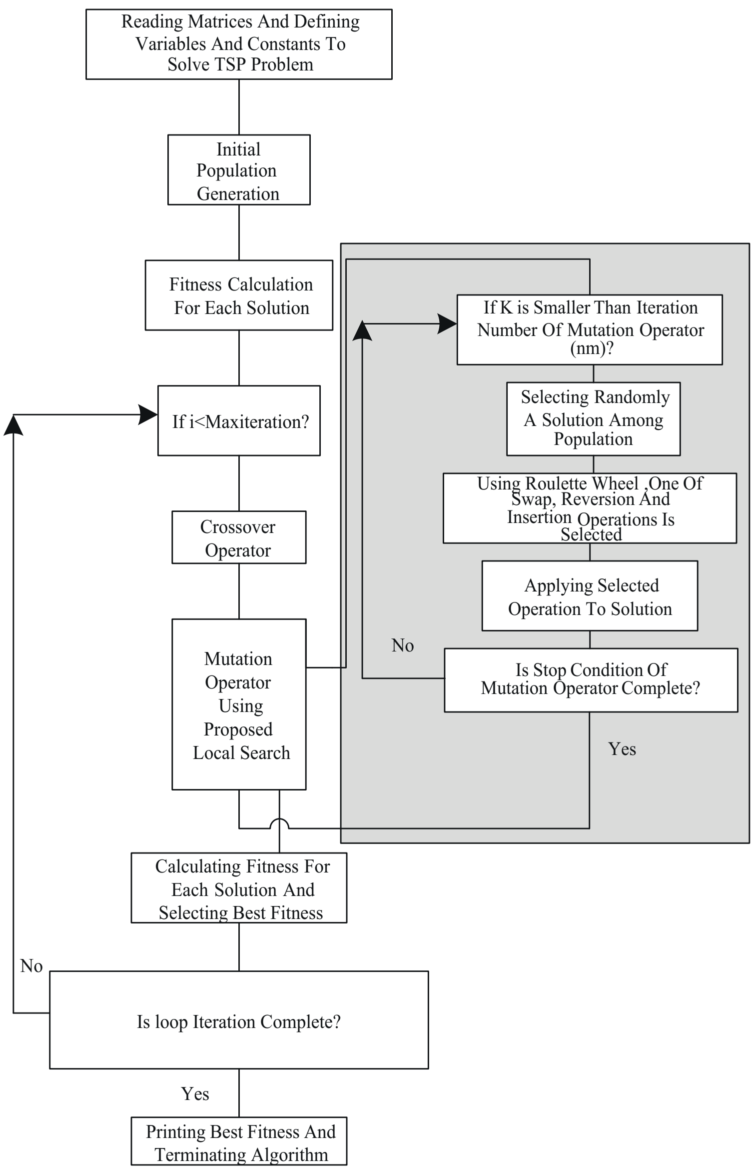

Figure 1. Flowchart of proposed memetic GA.

In a memetic GA, the proposed local search is used in the mutation operator in which only two genes are selected to change the basic GA. Their places are swapped and only this swap operation is applied. In the proposed local search, insertion and reversion operations are used in addition to the swap operation in order to find more diverse solutions. At each mutation operation, using a roulette wheel, one of the three is selected.

In the current study, the probability of selection of each method on the roulette wheel are: ProSwap = 0.3, ProInversion = 0.45 and ProInsersion = 0.25. The probabilities have been adjusted because, when they are considered in this way, after averaging the results of the proposed method they are much better than when the chances are equal. In addition, the evaluation of the results shows that the reversion method presents more proper changes than the other two methods (swap and insersion methods).

5.3. Genetic algorithm with a complex network

The proposed method combines basic GA and the complex network to solve the TSP by using a complex network in places where decision-making and solution selection are required. In the evolutionary GA, the complex network must be investigated in order to select solutions for moving to the next generation. In this case, a solution is selected based on the network status. In the proposed method, the node degree is measured and used for complex network analysis. Each network has nodes and edges and, in the complex network, nodes represent solutions. Edges represent the relationship between nodes. The focus of the current study is on the GA and, in this case, the relationship is the parent-child relationship between solutions. When forming a complex network, a data structure is required that maintains the network information, including the number of nodes and degree of each node. This data structure is referred to as "POP" and is used to maintain the parameters required for the initial population and information about the population which will be generated in the succeeding generations. Table 1 shows the solution number, solution path, solution cost and degree of each node and their applications.

Table 1. Pop data structure.

Data name	Data type	Application
Solution number	Integer	Each solution is named with a new number called the solution number.
Solution path (position)	An array of n*1 where n is the number of cities	It represents the route from the origin to all the nodes.
Cost	Decimal number	It represents the fitness of a route.
Node degree (degree)	Integer	It represents the degree of each node.

5.3.1. Selection operations with a complex network

Using random selection and nPOP initial population generation, the parameters of the POP data structure can be filled. For each new solution, a number is added to the solution number (number of solutions) and the information about that solution is written in the relevant row of the structure. When a solution is selected, its degree is set to one and the solution path is placed in the position array. The fitness value of the path is calculated using the cost function and the crossover and mutation operations are performed as explained in the next sub section. This operation should be repeated nPOP times, which is equivalent to the size of the initial population.

5.3.2. Crossover operation with complex network

The three parent selection methods discussed below were considered in the current study.

Cost-based roulette wheel: The basic GA is based on this method. When a parent is selected from a generation, the solution with the best fitness will have a greater chance of being selected in the roulette wheel and of moving forward to the next generation. Degree-based roulette wheel: This method selects the solution with the highest degree from the specific generation. It is used for a complex network and the degree-based complex network is used to apply operations to the GA. Mixed roulette wheel: In this method, solution selection is based on both the cost-based and degree-based roulette wheels in order to increase the performance of selections of the solutions. The fitness of two solution selections is compared and the solution with the highest fitness is transferred to the next generation. The flowchart of the GA with a complex network is shown in Figure 2.

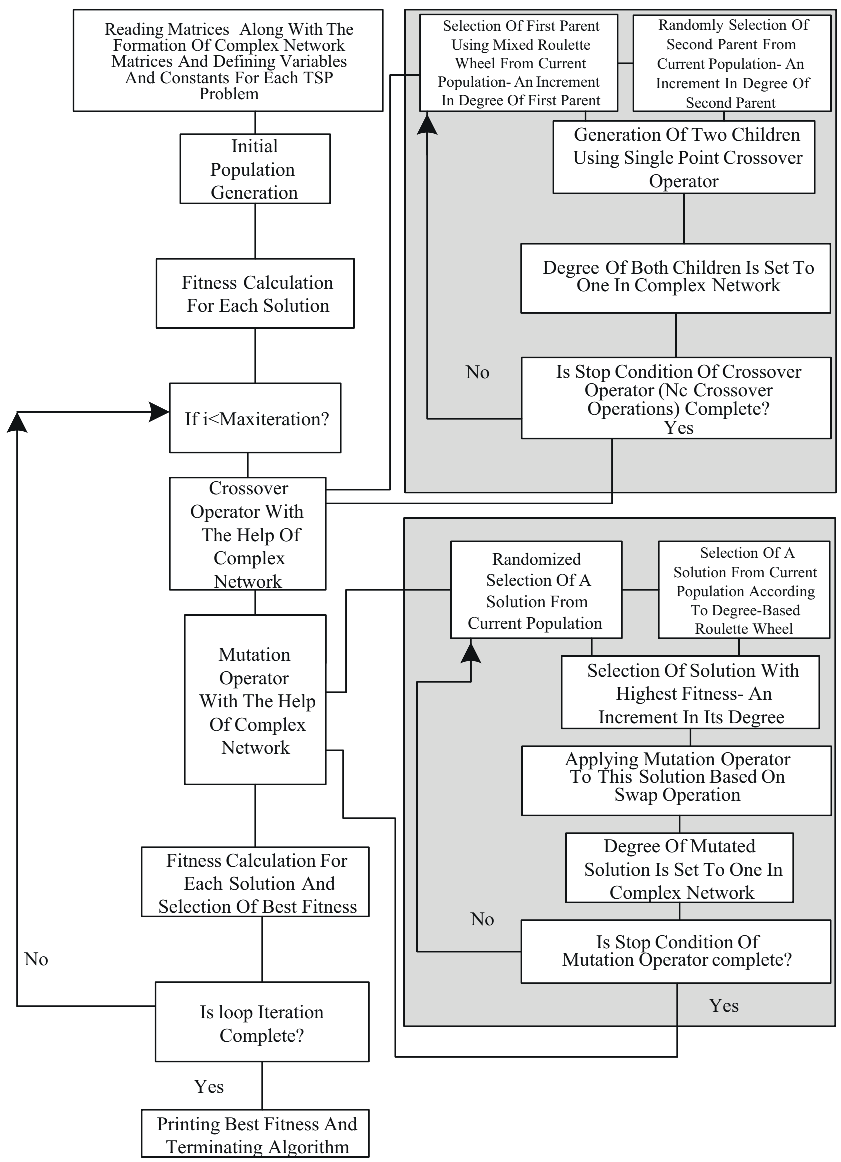

Figure 2. Flowchart of GA with complex network.

Unlike the basic GA, where two parents that are randomly selected based on the solution with the highest fitness have higher chances of being selected in the roulette wheel from the initial population or the current generation to generate a new generation, the mixed roulette wheel is used to select the first parent. The second parent is also randomly selected from the individuals of the current generation. This operation increases the degrees of both parents in the complex network. In the first implementation, because all node degrees are equal to one, the chance of selecting any particular solution as a parent is the same for all members of the population. The child selected will either be repetitive or not repetitive. Repetitiveness can be determined by comparing the solutions in the POP structure. If the child is not repetitive, it is considered to be a new child. In this method for both types of children is to set the degree of the child to one. In the mutation operator, the selected solution is randomly chosen from the current population and a solution also is selected using the degree-based roulette wheel. The costs of both solutions are then compared and the solution with the lowest cost is selected as the one in which the mutation should be implemented. In a complex network, the degree of the solution will increase. The mutation operation is performed on the solution based using only the swap operation to make changes and the node degree of the mutated solution is set to one. The average of this method is better than for the basic GA, but it performs much more weakly than the memetic GA.

5.4. Genetic algorithm with a complex network and proposed local search

This is the form of the third method. In addition to applying a degree-based complex network to the GA, the operations of swap, reversion and insertion have been used in the mutation operator in order to apply changes to the solution. The local search proposed in the second method has been applied to the mutation part and the results are similar to those of the memetic GA method, although the fitness value is slightly lower. The drawback of this method is its early convergence to a solution with a higher degree. Figure 3 is a pseudo-code of a mutation operator with the help complex network and the proposed local search.

```
Start: Mutation Operator
Do
        Randomized Selection of a Solution from Current Population(RS)
        Selection of a solution from Current Population According to Degree-based Roulette Wheel(DS)
        Between RS and DS Solutions, Selection of Solution with Best Fitness
        Degree Selected Solution = Degree Selected Solution + 1
        Applying Mutation Operator to This Solution based on Swap, Reversion and Insertion Operations
        Degree of Mutated Solution = 1
While(Stop Condition of Mutation Operator is Complete)
End
```

Figure 3. Pseudo-code of mutation operator with the help of complex network and local search.

5.5. Genetic algorithm with a complex network and proposed local search with degree limitation

This method is a variation of the fourth method in which two changes have been made. At the crossover stage, the mixed roulette wheel has been used to select parents which have lower fitness from the cost and degree based roulette wheels. In order to prevent early convergence to the node with highest degree, a maximum value is set for the degree value of each node in the complex network. This is referred to as “maxrank” which is equal to 69. This prevents the algorithm from converging to a node with a high degree. Figure 4 is a flowchart of a GA with a complex network and proposed local search with degree limitation.

5.6. Genetic algorithm with complex network and proposed local search with degree limitation and population diversity

This method is a variation of the fifth method in which the future population improvement method has been used to prevent early convergence. At the end of the crossover and mutation operations to produce the population

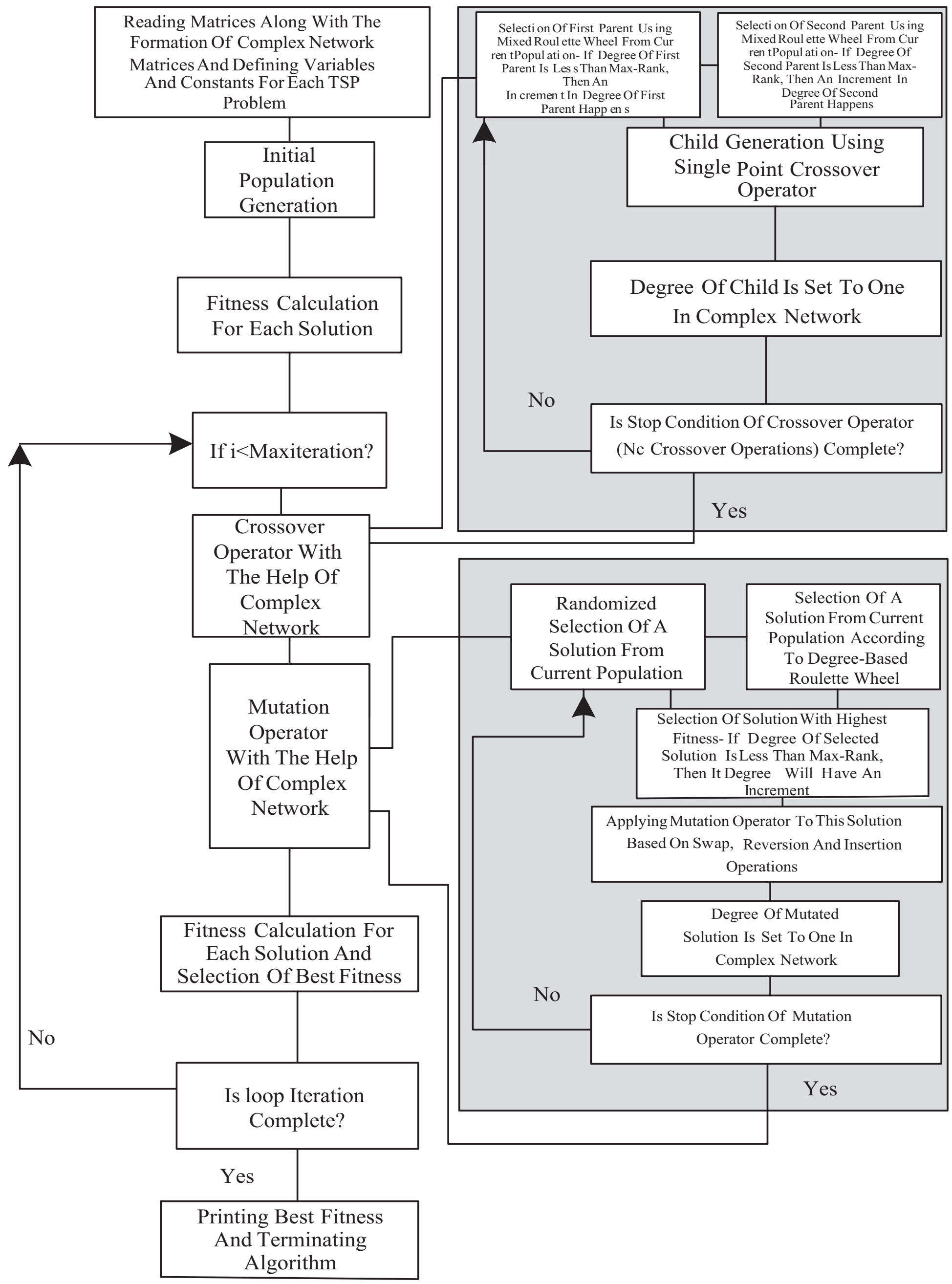

Figure 4. Flowchart of GA with complex network and proposed local search with degree limitation.

of next generation, instead of selecting all solutions with the best fitness, 25% of options are randomly selected that are twice the size of the current population in the memory and 75% are selected from among the fittest. This provides diversity in the population and decreases the early convergence in the algorithm. Figure 5 is a pseudo-code of GA along with complex network and proposed local search with degree limitation and population diversity.

01. Start: Sixth Method Algorithm
02. Read: Data
03. Initial Population Generation
04. Fitness Calculation for each Solution
05. DO
06. Crossover Operator with the Help of Complex Network and Local Search
07. Mutation Operator with the Help of Complex Network and Local Search
08. For Each Solution **Do**
09. Fitness Calculation
10. Selection Of Best Fitness
11. End For
12. Next Generation Formation of Solutions with 25% of Twice the Size of Population from Solutions in addition to 75% of Solutions with Best Fitness
13. While (Loop Iteration is Terminate)
14. End

Figure 5. Pseudo-code of GA along with complex network and proposed local search with degree limitation and population diversity.

5.7. Solving TSP using a complex network with degree measure and three-hub network formation with proposed local search

In this method, a complex network with degree measure has been used. Three of the nodes grow in degree and the remaining nodes are selected as children and added to one of the three nodes. During the generation of parent and child, changes have been made to the crossover and mutation operators. Figure 6 is a pseudo-code of a three-hub complex network algorithm with the proposed local search.

The first three solutions with the lowest costs among the initial population are selected as the fixed parents (hubs) in the crossover and mutation operators. The first and second parents of the crossover operator and the selected solution in the mutation operator are always considered to be the three fixed parents. For child generation, two parents are selected for the crossover operator using the degree-based roulette wheel, but their degrees are not added to the complex network because the base is the only connection between all children and the three parents as hubs. The result of using the crossover operator for the two parents is two children for which the degrees of the children are set to one in the complex network. The solution number of the selected children is added to the list of nodes connected to the hubs. The degree of solution which is selected using the degree-based roulette wheel in the mutation operator does not change in the complex network. Rather, changes are made to the solution using the swap, reversion and insertion operations of the local search. The degree of the changed solution node is set to one in the complex network. Only the degree of the node selected as the third fixed hub in the first stage increases. The first and second parents in the crossover operator are always the fixed hubs selected from the initial population. Therefore, the solution upon which the mutation operator is performed is always a fixed parent of the initial population. Only the degrees of the three hub nodes increase and the degrees of the nodes generated as children using the crossover or mutation are set to one. The proposed local search also is used in the mutation stage.

6. Results

The seven problems of the TSPLib set has been compared using the GA, memetic GA and complex network. The parameters have been set according initial population = 150, problem iteration number = 5000 and seven different TSPLib problems with the names burma14, ftv33, ft53, ST70, gr137, ch150 and d198.

```
Start: Seventh Method Algorithm
Read:Data
Initial Population Generation
Fitness Calculation for each Solution
Determination of Three Solution with Best Fitness as First and Second Parents in Crossover Operator and
Three Solution as a Solution in Which Mutation Operation Will be Performed
DO
    DO//Crossover Operator with the Help of Complex Network and Local Search
        Degree First Hub = Degree First Hub + 1
        Selection of First Parent with Degree-based Roulette Wheel
        Degree Second Hub = Degree Second Hub + 1
        Selection Of Second Parent With Degree-based Roulette Wheel
        Generation of Two Children Using Single Point Crossover Operator
        Degree First Child = 1
        Degree Second Child = 1
        First Child is Connected to First Hub and Second Child is Connected to Second Hub
    While (Stop Condition of Crossover Operator is Complete)
    DO//Mutation Operator with the Help of Complex Network and Local Search
        Degree Three Hub = Degree Three Hub + 1
        Selection of a Parent with Degree-based Roulette Wheel
        Applying Mutation Operator to This Solution based on Swap, Reversion and Insertion Operations
        Degree Mutated Solution = 1
        Connection of This Child to Hub of Mutation Operator
    While (Stop Condition of Mutation Operator is Complete)
For Each Solution Do
    Fitness Calculation
    Selection of Best Fitness
End For
While (Loop Iteration is Complete)
Printing Best Fitness
End
```

Figure 6. Pseudo-code of a three-hub complex network algorithm with the proposed local search.

To allow comparison of the results, each method has been successively run 30 times. The rows in Table 2 show the best, worst, and average fitness, percentage error and average run time for each TSP problem with 30 runs. The basic GA method (section 5.1) obtained the worst results among the proposed methods. The memetic GA method (section 5.2) performed relatively well because it uses the proposed local search. The GA with a complex network and proposed local search with degree limitation (section 5.5) obtains a well result by controlling convergence. The best performance was recorded by solving the TSP using a complex network with degree measure and three-hub network formation with the proposed local search (section 5.7).

In order to evaluate the error rate of the proposed methods with the best results in TSPLib, a criterion called percentage error can be calculated as Formula 3:

$$PercentageError = \frac{ProposedSoluFit - BKS}{BKS} * 100, \tag{3}$$

where proposed solution is the fitness rate of the proposed solution and BKS is the best solution in the TSPLib database. The value that is closest to zero is the best result. In this paper, seven TSPLib problems were solved by the proposed methods that in most cases the seventh method had the best fitness and the first method had the worst fitness. The percentage error for the seven different TSPLib problems was shown in Figure 7.

Table 2. Comparison of the results of different algorithms.

Problem name	Metrices	First ■ method	Second ■ method	Third ■ method	Fourth ■ method	Fifth ■ method	Sixth ■ method	Seventh ■ method
burma14	Best fitness	31	31	31	31	31	31	31
	Worst fitness	31	31	31	31	31	31	31
	Average fitness	31	31	31	31	31	31	31
	Standard deviation	0	0	0	0	0	0	0
	Percentage error	–7.08	–7.08	–7.08	–7.08	–7.08	–7.08	–7.08
	Average run time	152	152	341	334	336	343	960
ftv33	Best fitness	1565	1445	1520	1399	1445	1488	1329
	Worst fitness	1923	1596	1841	1797	1578	1587	1452
	Average fitness	1790	1512	1704	1575	1532	1535	1403
	Standard deviation	123.06	54.79	88.19	168.63	45.52	36.51	42.83
	Percentage error	21.7	12.36	18.2	8.79	13.06	15.71	3.34
	Average run time	152	153	343	336	338	343	963
ft53	Best fitness	9984	7869	9200	7489	7645	7809	7424
	Worst fitness	10955	8929	11359	9528	16585	10303	8644
	Average fitness	10528	8510	10388	8647	10030	8827	8040
	Standard deviation	327.06	266.20	721.65	675.61	3004.08	756.11	428.73
	Percentage error	44.59	13.96	33.24	8.46	10.72	13.09	7.52
	Average run time	155	155	360	349	348	350	969
ST70	Best fitness	936	683	927	686	685	692	677
	Worst fitness	1225	745	1188	747	743	752	741
	Average fitness	1077	712	1055	713	706	716	701
	Standard deviation	73.55	15.87	60.69	16.26	16.7	15.06	13.44
	Percentage error	38.66	1.15	37.31	1.69	1.47	2.54	0.31
	Average run time	156	155	360	355	355	368	972
gr137	Best fitness	1238	719	1142	717	723	717	712
	Worst fitness	1651	775	1552	784	776	782	789
	Average fitness	1392	752	1365	747	751	748	744
	Standard deviation	106.39	12.94	103.02	16.33	14.27	14.3	16.67
	Percentage error	77.27	7.72	63.45	2.67	3.48	2.67	1.87
	Average run time	158	159	359	390	377	369	986
ch150	Best fitness	11026	6778	11160	6795	6779	6802	6668
	Worst fitness	14393	7195	14723	7369	7528	7334	7293
	Average fitness	12815	7023	12913	7044	7057	7066	6940
	Standard deviation	843.98	106.19	950.24	151.42	180.74	121.59	156.36
	Percentage error	68.9	3.83	70.96	4.09	3.85	4.2	1.87
	Average run time	162	161	364	382	376	387	990
d198	Best fitness	31549	16044	33501	16116	16183	16090	16079
	Worst fitness	50293	16704	47781	16815	16815	16650	16908
	Average fitness	40187	16402	39949	16434	16435	16397	16431
	Standard deviation	4464.22	150.48	4136	158.44	152.62	156.19	196.61
	Percentage error	99.93	1.67	112.3	2.13	2.55	1.96	1.89
	Average run time	173	177	369	380	384	394	1011

For example, for the ST70 problem, The best percentage error value of 0.31% was recorded by the method that solved the TSP using a complex network with degree measure and three-hub network formation with the proposed local search. The worst error was recorded by the basic GA at 38.66%.

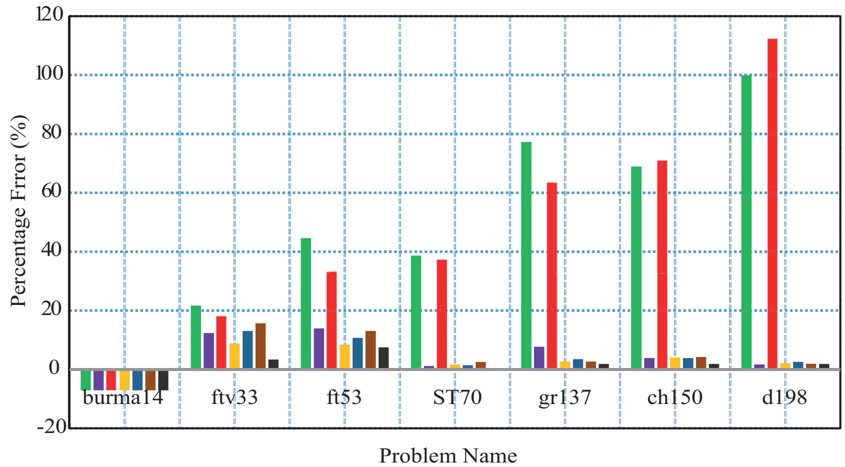

Figure 7. Comparison of the percentage error of different algorithms for the seven different TSPLib problems. ■ First method, ■ Second method, ■ Third method, ■ Fourth method, ■ Fifth method, ■ Sixth method, and ■ Seventh method.

In addition to the fitness, the average of the proposed methods for solving TSPLib database problems was also evaluated. The algorithms were implemented in a system with a CPU Core i5 2.5 GHz with 4 GB RAM. The average run time of the algorithms indicated their scalability. For example, when the seventh method was used for problems with sizes 14 and 198, they lasted 960 and 1011 s, respectively. The reason that the seventh method took more time than the other methods could be traced in the formation of complex networks and the maintenance of interconnected nodes. The t-test was also used to evaluate the proposed methods. The statistical results of the t-test with confidence interval at 95% significance level led to the significance level value of zero. This result illustrated the significant difference between the proposed algorithms and the basic genetic algorithm to obtain the fitness of traveling different cities in the travelling salesman problem. Figure 8 depicts the convergence curve of the seven best-runing algorithms for four different TSPLib problems with the size of cities as 53,70,137, and 198.

7. Conclusion and future works

The current study used a GA to solve the TSP and then changed the GA to a memetic GA using the proposed local search. In the real world, the relationship between generations is made with the help of genes, but this problem has not been considered in the basic GA. A complex network has been developed to provide this relationship between generations, parents and children. In this network, the hereditary and generational relationship between chromosomes has been used and the node degree measure has been used to evaluate the network. At each stage, decisions are made about selecting a chromosome by evaluating the complex network. The node with the highest degree has the greatest chance of being selected. Several scenarios have been considered to investigate the effect of complex networks on solving the TSP. The results revealed better fitness using the complex network and the memetic GA than the basic algorithm and the memetic GA for solving the TSP. Also, the average run time of the algorithms showed their scalability. The performance of the proposed

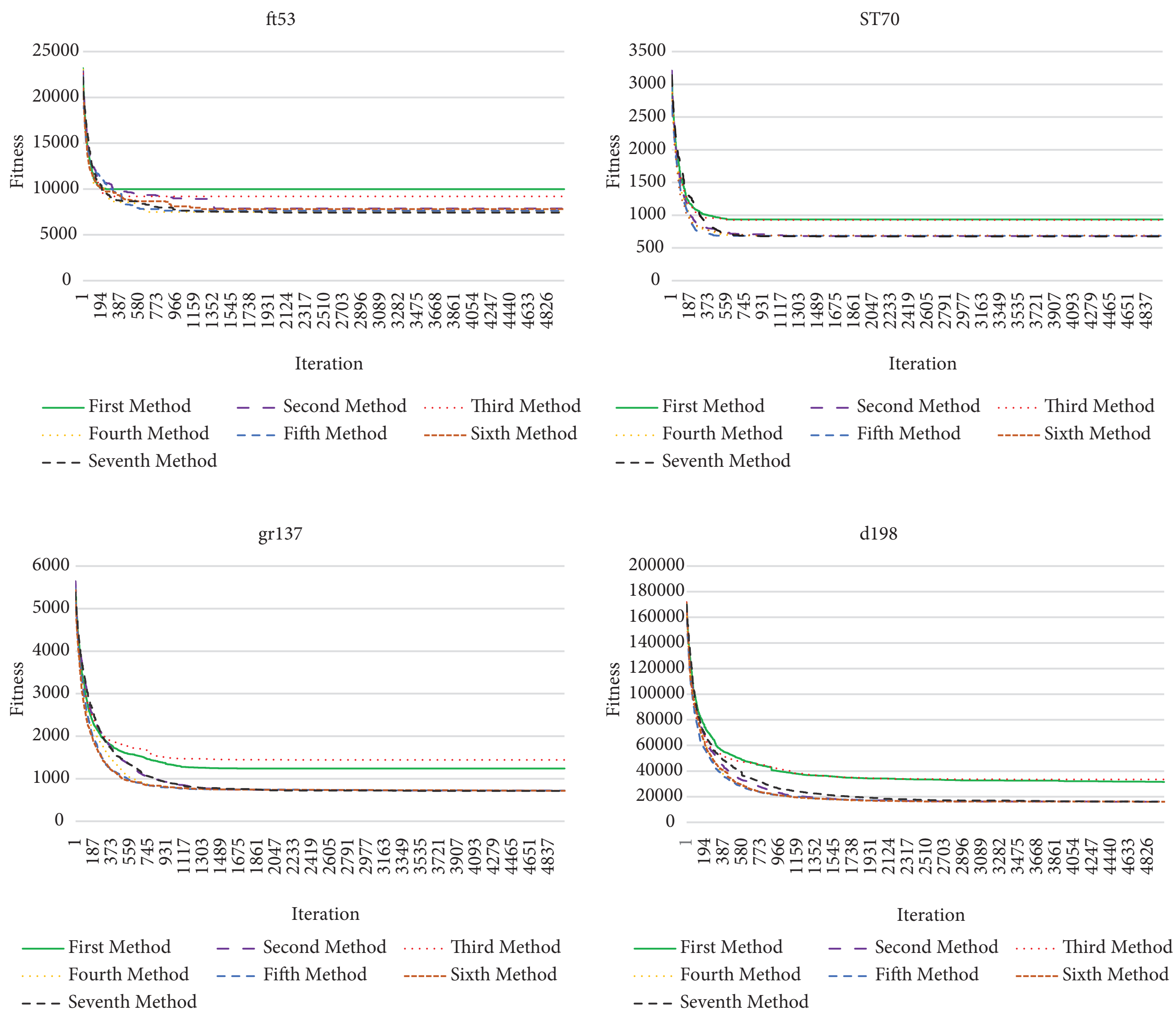

Figure 8. The convergence curve of the seven best-runing algorithms for four different TSPLib problems with the size of cities as 53,70,137, and 198.

memetic algorithms could be improved by applying other methods to change the parameters used in the GA. Memetic methods such as tabu search and simulated annealing are other possibilities. Measures such as the clustering coefficient, degree centrality and closeness centrality can also be considered for analysis of the network in a complex network algorithm.

References

[1] Fuentes GEA, Gress ESH, Mora JCS, Marín JM. Solution to travelling salesman problem by clusters and a modified multi-restart iterated local search metaheuristic. PLoS One 2018; 13 (8): 1-20. doi: 10.1371/journal.pone.0201868

[2] Nagarani B, Nesamony J. Performance enhancement of photovoltaic system using genetic algorithm-based maximum power point tracking. Turkish Journal of Electrical Engineering & Computer Sciences 2019; 27 (4): 3015-3025. doi: 10.3906/elk-1801-189

[3] Peker M, Sen B, Kumru PY. An efficient solving problem: the ant colony system having parameters optimized by the Taguchi method. Turkish Journal of Electrical Engineering & Computer Sciences 2013; 21 (1): 2015-2036. doi: 10.3906/elk-1109-44

[4] Hekim M, Comert O, Adem K. A hybrid model based on the convolutional neural network model and artificial bee colony or particle swarm optimization-based iterative thresholding for the detection of bruised apples. Turkish Journal of Electrical Engineering & Computer Sciences 2020; 28 (1): 61-79. doi: 10.3906/elk-1904-180

[5] Guvenc U, Yigit T, Isik AH, Akkaya I. Performance analysis of biogeography-based optimization for automatic voltage regulator system. Turkish Journal of Electrical Engineering & Computer Sciences 2016; 24 (3): 1150-1162. doi: 10.3906/elk-1311-111

[6] Kalinli A. Simulated annealing algorithm-based elman network for dynamic system identification. Turkish Journal of Electrical Engineering & Computer Sciences 2012; 20 (4): 1150-1162. doi: 10.3906/elk-1012-942

[7] Jemai M, Dimassi S, Ouini B, Mtibaa A. A metaheuristic based on the tabu search for hardware-software partitioning. Turkish Journal of Electrical Engineering & Computer Sciences 2017; 25 (2): 901-912. doi: 10.3906/elk-1501-64

[8] Rostami N. Particle swarm optimization approach to optimal design of an AFPM traction machine for different driving conditions. Turkish Journal of Electrical Engineering & Computer Sciences 2019; 27 (4): 3234 -3246. doi: 10.3906/elk-1805-176

[9] Mollakhalili Meybodi MR, Meybodi MR. Extended distributed learning automata an automata-based framework for solving stochastic graph optimization problems. Applied Intelligence 2014; 41 (2): 923-940. doi: 10.1007/s10489-014-0577-2

[10] Zhang JW, Si WJ. Improved enhanced self-tentative PSO algorithm for TSP. In: IEEE 2010 Sixth International Conference on Natural Computation (ICNC) Conference; Yantai, China; 2010. pp. 2638-2641.

[11] Zhou Tao Z. TSP problem solution based on improved genetic algorithm. In: IEEE 2008 Fourth International Conference on Natural Computation Conference; Jinan, China; 2008. pp. 686-690.

[12] Zhang L, Xiao C, Fei T. Improved ant colony optimization algorithm based on RNA computing. Automatic Control and Computer Sciences 2017; 51 (5): 366-375. doi: 10.3103/s0146411617050108

[13] Buriol LS, Moscato P, França PM. A new memetic algorithm for the asymmetric traveling salesman problem. Journal of Heuristics 2004; 10 (5): 483-506. doi: 10.1023/b:heur.0000045321.59202.52

[14] Rezapoor Mirsaleh M, Meybodi MR. Balancing exploration and exploitation in memetic algorithms: a learning automata approach. Computational Intelligence 2018; 34 (1): 282-309. doi: 10.1111/coin.12148

[15] Gao Y, Du W, Yan G. Selectively-informed particle swarm optimization. Scientific Reports 2015; 5 (1): 1-7. doi: 10.1038/srep09295

[16] Matsushita H, Nishio Y, Saito T. Particle swarm optimization with novel concept of complex network. In: 2010 International Symposium on Nonlinear Theory and Its Applications Conference; Krakow, Poland; 2010. pp. 197-200.

[17] Watts DJ, Strogatz SH. Collective dynamics of small-world networks. Nature 1998; 393 (6684): 440-442. doi: 10.1038/30918

[18] Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang D-U. Complex networks: structure and dynamics. Physics Reports 2006; 424 (4-5): 175-308. doi: 10.1016/j.physrep.2005.10.009

[19] Mo H, Xu L. Biogeography based optimization for traveling salesman problem. In: IEEE 2010 Sixth International Conference on Natural Computation (ICNC) Conference; Yantai, China; 2010. pp. 3143-3147.

[20] Simon D. Biogeography-based optimization. Transactions on Evolutionary Computation 2008; 12 (6): 702-713. doi: 10.1109/TEVC.2008.919004

[21] Pham DT, Huynh TTB, Dang MC. An improving of migration operator in biogeography-based optimization for solving traveling salesman problem. In: IEEE 2017 International Conference on Computational Intelligence and Cybernetics Conference; Makassar, Indonesia; 2016. pp. 33-40.

[22] Li M, Du W, Nian F. An adaptive particle swarm optimization algorithm based on directed weighted complex network. Mathematical Problems in Engineering 2014; 2014 (2): 1-7. doi: 10.1155/2014/434972

[23] Liu C, Du1 WB, Wang WX. Particle swarm optimization with scale-free interactions. PLoS One 2014; 9 (5): 1-8. doi: 10.1371/journal.pone.0097822

[24] Contreras-Bolton C, Parada V. automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. PLoS One 2015; 10 (9): 1-25. doi: 10.1371/journal.pone.0137724

[25] Thanh PD, Binh HTT, Lam BT. New mechanism of combination crossover operators in genetic algorithm for solving the traveling salesman problem. Knowledge and Systems Engineering 2015; 326 (1): 367-379. doi: 0.1007/978-3-319-11680-8-29

[26] Sabry D, Abdel-Moetty Asmaa M, Heakil O. Enhanced traveling salesman problem solving using genetic algorithm technique with modified sequential constructive crossover operator. IJCSNS International Journal of Computer Science and Network Security 2012; 12 (6): 134-139.

[27] Xu Q, Wang N, Wang L. Enhancing performance of oppositional BBO using the current optimum (COOBBO) for TSP problems. International Journal of Intelligent Computing and Cybernetics 2016; 9 (2): 144-164. doi: 10.1108/ijicc-03-2016-0015

[28] Xu Q, Wang L. A novel oppositional biogeography-based ptimization for combinatorial problems. In: IEEE 2014 10th International Conference on Natural Computation (ICNC) Conference; Xiamen, China; 2014. pp. 414-420.

[29] Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. In: IEEE 2005 International Conference on Computational Intelligence for Modelling International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06) Conference; Vienna, Austria; 2005; pp. 695-701.

[30] Mitchell M. Complex systems: network thinking. Artificial Intelligence 2006; 170 (2006): 1194–1212. doi: 10.1016/j.artint.2006.10.002

[31] Zelinka I, Chen G. Evolutionary algorithms. In: Zelinka I, Chen G (editors). Swarm Dynamics and Complex Networks Methodology Perspectives and Implementation. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2018.