Esra'a Alkafaween - Ahmad B. A. Hassanat

IMPROVING TSP SOLUTIONS USING GA WITH A NEW HYBRID MUTATION BASED ON KNOWLEDGE AND RANDOMNESS

Genetic algorithm (GA) is an efficient tool for solving optimization problems by evolving solutions, as it mimics the Darwinian theory of natural evolution. The mutation operator is one of the key success factors in GA, as it is considered the exploration operator of GA.

Various mutation operators exist to solve hard combinatorial problems such as the TSP. In this paper, we propose a hybrid mutation operator called "IRGIBNNM", this mutation is a combination of two existing mutations; a knowledge-based mutation, and a random-based mutation. We also improve the existing "select best mutation" strategy using the proposed mutation.

We conducted several experiments on twelve benchmark Symmetric traveling salesman problem (STSP) instances. The results of our experiments show the efficiency of the proposed mutation, particularly when we use it with some other mutations.

Keyword: *knowledge-based mutation, inversion mutation, slide mutation, RGIBNNM, SBM*

1 Introduction

1.1 Travelling salesman problem (TSP)

TSP is considered as one of the combinatorial optimization problems [1], that is easy to describe but difficult to solve, and it is classified among the problems that are not solved in polynomial time; i.e. it belongs to the NP-hard problem [2].

A solution of TSP aims at finding the shortest path (tour) through a set of nodes (starting from a node N and finishing at the same node) so that each node is visited only once [3].

The classic problem of a traveling salesman is an active and attractive field of research because of its simple formulation [2], and it was proved to be NP-complete problem, since no one found any effective way to solve an NP problem of a large size, in addition, many problems in the world can be modeled by TSP [4].

The TSP is classified into:

1. Symmetric traveling salesman problem (STSP): The cost (distance) between any two cities in both directions is the same (undirected graph), i.e. the distance from *city1* to *city2* is the same as the distance from *city2* to *city1*. There are (N-1)! /2 possible solutions for N cities.
2. Asymmetric travelling salesman problem (ATSP): The cost between any two cities in both directions is not the same. There are (N-1)! possible solutions for N cities [5].

TSPs are used in various applications, including: job sequencing, computer wiring, crystallography, wallpaper cutting, dartboard design, hole punching, overhauling gas turbine, etc. [6].

Over the years various techniques have been suggested to solve the TSP, such as genetic algorithm (GA) [7-8], hill climbing [9], nearest neighbor and minimum spanning tree algorithms [10], simulated annealing [11], ant colony [9], tabu search [12], particle swarm [13], elastic nets [14], neural networks [15], etc. Genetic algorithms are one of the algorithms that extensively applied to solve the TSP [16].

1.2 Genetic algorithm (GA)

GA is an optimization algorithm [17] that is classified as global search heuristic; it is one of the categories that form the family of the evolutionary algorithms, which mimics the principles of natural evolution [18]. GA is a population-based search algorithm, as in each generation, a new population is generated by repeating three basic operations on the population, namely, selection, crossover, and mutation [19]. GA has been used extensively in many fields, such as computer networks [20], speech recognition [21], image processing [22], software engineering [23], etc.

A simple GA algorithm is described as follows [16]:

Step1: Create a random population of potential solutions [24] consisting of n individuals (initial populations): The first phase of any GA is initial population seeding. It generates a first population randomly or by heuristic initialization as

---

Esra'a Alkafaween[1,*], Ahmad B. A. Hassanat[1,2,3]
[1] IT Department, Mutah University, Karak, Jordan
[2] Computer Science Department, Community College, University of Tabuk, Saudi Arabia
[3] Industrial Innovation and Robotics Center, University of Tabuk, Saudi Arabia
*E-mail of corresponding author: esrakafaween86@gmail.com

input for the GA [25], such as: random, nearest neighbor, k-means clustering and initialization mechanism based on regression techniques [26].

Step2: Evaluate the fitness value f(x) of each individual, x, in the population.

Step3: Repeat the following three steps to create a new population until completion of the new population.

Step4: Selection phase: This is the process of choosing the best parents of the current generation in the community for mating, such as: roulette wheel, elitism and tournament.

Step5: Apply crossover with a certain ratio to create offspring: This process takes two parents (chromosomes) to create a new offspring by switching segments of the parent genes. It is more likely that the new offspring (children) will contain good parts of their parents, and consequently perform better as compared to their ancestors. There are many types of crossover, such as: modified crossover, uniform crossover and arithmetic crossover.

Step6: Apply mutation with a certain ratio: This is where there is a change or a switch between specific genes within a single chromosome to create chromosomes that provide new solutions for the next generation, with the aim of obtaining the best possible solutions, and thus introduce a certain level of diversity to the population, and as a result this also does not fall into the local optimum [27]. There are many types of mutation such as: exchange mutation, scramble mutation, insert mutation and inversion mutation.

Step7: The previous operations are repeated until the completion criterion is met.

The performance of the GA is affected by several key factors, such as the population size, the selection's strategy, the mutation operator used, the crossover operator used and the coding scheme [28-30]. In this paper, we focus on the mutation operator.

Mutation operator plays an important role in the GA, where it helps to stimulate the GA to explore new areas in the search space [19]. It is an effective mechanism for preserving the diversity of individuals [28], where mutation provides variation in the population through random changes of individuals [29]. And therefore, overcoming the so-called premature convergence [31], also preventing the loss of genetic material [32].

In this paper, we propose a hybrid mutation operator called inversion RGIBNNM (IRGIBNNM) to provide an efficient solution to TSP, we use simple GA with mutations only; there is no other variable/parameter that controls the workflow of such a simple GA, as we want to examine the strength of the proposed mutation apart from the effect of other parameters; we compare the performance of this mutation with the performances of three existing mutations, and we used it with two other mutations to form a multi-mutations GA. The comparisons are made on symmetric TSP instances.

The organization of this paper is as follows. In Section 2 we present some of the related work. In section 3, we present the proposed mutation, the existing two mutations and the mutation strategy. In Section 4 we describe the experiments conducted, and discuss them. The conclusion is written in Section 5.

2 Background

Over the years, researchers have suggested several types of mutations to be used in various types of encoding, including: flip mutation, creep mutation and insert mutation [33], gaussian mutation, exchange mutation [34], displacement mutation [35], uniform mutation [1], inversion mutation [36] and some other types.

Louis and Tang proposed a new mutation called greedy-swap mutation, where two cities are chosen randomly from the same chromosome, and switching between them if the length of the new tour obtained is “better” (shorter) than the previous ones [37].

Potvin [2] and Larranaga et al. [8] presented a review of representing the TSP, explaining the advantages and disadvantages of different mutation operators. Soni and Kumar studied many types of mutations that solve the problem of travelling salesmen, including interchanging mutation, reversing mutation and scramble mutation [1]. Otman and Jaafar used reverse sequence mutation (RSM) and several types of crossover to solve the TSP [32]. Korejo et al, introduced a directed mutation (DM), this method used the statistical information provided by the current population to explore the promising areas in the search space [19].

Having such a large number of mutations, the problem becomes which mutation to use? As the problem lies in choosing the appropriate mutation. To answer this question, several researchers have developed new types of GAs that use more than one mutation at the same time [38-41].

Katayama et al. presented a promising GA for TSP, called a hybrid mutation genetic algorithm (HMGA), which employed a local search algorithm called stochastic hill climbing (SHC), in order to avoid falling into the local optima [42].

Hong et al. proposed a new GA, called dynamic genetic algorithm (DGA) in order to choose the appropriate mutation and crossover operators and their ratios automatically, this algorithm use more than one mutation at the same time, such as: uniform crossover, (0,1) change, inversion, bit-change and swapping [43].

Hassanat et al. proposed 10 types of knowledge-based mutations; the most promising one is called “random gene inserted beside nearest neighbor mutation” (RGIBNNM). In addition, they proposed two selection strategies for the mutation operators called: “select the best mutation” (SBM) and “select any mutation” (SAM). They applied all mutations and strategies on several TSP instances [38].

Regardless the extensive research in this domain, there is no one mutation ideally suited for all TSP instances. Since no one method found in the literature that guarantees an optimal solution for any TSP instance. Therefore, there is still room for improvement in this domain.

3 The proposed method

In this section, we explain some of the existing mutation operators that are proposed for the permutation coded GA; these include slide mutation [44], inversion mutation [36] and RGIBNNM [38]. Moreover, we explore the strategy of choosing the best mutation; the SBM [38]. We also present the proposed hybrid mutation, which is nothing but a combination of the inversion mutation and the RGIBNNM, we call it IRGIBNNM.

3.1 Slide mutation

This mutation chooses two genes randomly, and then conveys the first to follow the second, and then shift the rest of the city, as depicted by *Example 1*.

Example 1: Consider the following TSP *tour C*:

*C=(**5 3 10 2 1 8 9 7 4 6**).*

If the third *gene 10* and the eighth *gene 7* are randomly selected, then the sub tour is:

*(**2 1 8 9 7**).*

The mutated tour will be:
*(Offspring)=(**5 3 10 1 8 9 7 2 4 6**).*

3.2 Inversion mutation

This mutation chooses two random genes, and then reverses the subset between them, as depicted by *Example 2*.

Example 2: Consider the following *tour C*:

*C=(**5 3 10 2 1 8 9 7 4 6**).*

If the third and eighth positions are randomly selected, then the sub tour is:

*(**2 1 8 9 7**),*

and then reversed to be:

*(**7 9 8 1 2**).*

The mutated tour will be:

*(Offspring)=(**5 3 10 7 9 8 1 2 4 6**).*

3.3 RGIBNNM mutation

This mutation is a knowledge-based operator designed especially for the TSP problem. However, it can be customized to fit some other problems. This operator uses the idea of the nearest neighbor cities, where this mutation selects a random gene (city), and finds its nearest city (Ncity), then swap the random city with one of the neighbors of the nearest city.

Example 3: suppose that the chromosome chosen for mutation is:

$B \rightarrow E \rightarrow C \rightarrow A \rightarrow D$,

as depicted in Figure 1. By applying RGIBNN:
Step 1: Suppose that the city which has been selected at random is *E*.
Step 2: Find the nearest city to the random city, which is *D* according to the distance table. This city is called Ncity.
Step 3: Now *E* is moved prior to *D*, and (*B* and *C*) are shifted to get a new chromosome

$B \rightarrow C \rightarrow A \rightarrow E \rightarrow D$.

3.4 The proposed IRGIBNNM

We propose a hybrid mutation called: IRGIBNNM.

In this mutation we combine two mutation operators, the inversion mutation and RGIBNNM.

The IRGIBNNM initially applies the inversion mutation on an individual, and then the RGIBNNM is applied to the resulting individual. Thus, the new offspring benefit from both mutations' characteristics, attempting to enhance the performance of both mutations, by increasing diversity in the search space, and therefore to provide better results. The IRGIBNNM is depicted by *Example 4*.

Example 4: Consider the following *tour C*:

C=(5 3 10 9 8 1 2 7 4) with cost=19,

as depicted in Figure 2. To apply IRGIBNNM:

Select two random genes, e.g. the third and seventh genes.

A=inversion mutation C.

The resulting offspring:

A=(5 3 10 2 1 8 9 7 4) with cost=18.2 (see Figure 3).

Apply RGIBNNM *A* as follows:

- Select a random gene from *A*, e.g. the random gene is the eighth gene, i.e. the *random city* is 7.
- Find the nearest city to the *random city 7*, which is *city 3* in this case.
- Get a random city around *city 3* in the range (±5) (e.g we determine the range of the number of cities close to the chosen city, so if we choose 3, we mean only the nearest 3 cities and then we choose one of them); e.g. *city 9*.

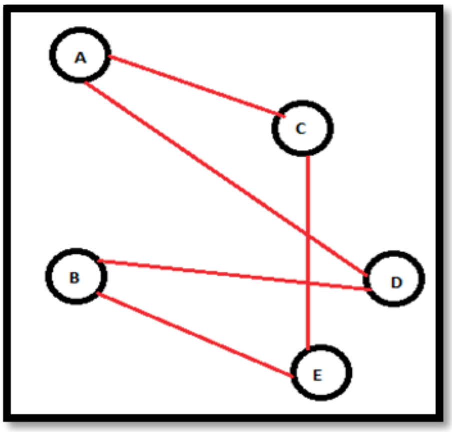

Figure 1 *Example of RGIBNNM*

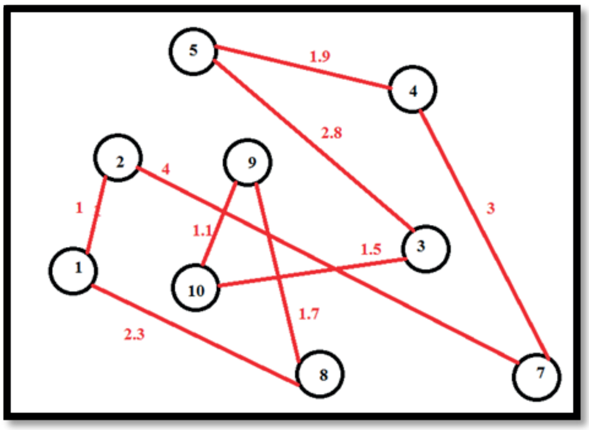

Figure 2 *Example of particular tour C with cost=19*

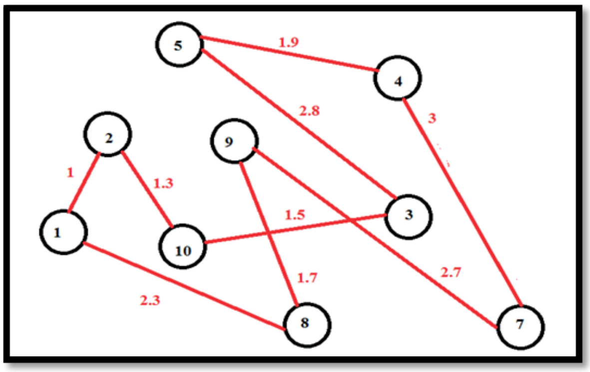

Figure 3 *Example of applying inversion mutation on C to get offspring A with cost=18.2*

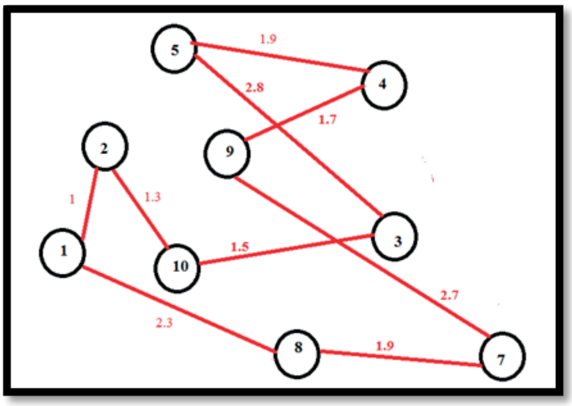

Figure 4 *Example of applying IRGIBNNM on A to get offspring with cost=17.1*

Apply the exchange mutation on chromosome *A* by swapping the *cities 7* and *9*, as shown in Figure 4. The final output offspring becomes:

Offspring=(5 3 10 2 1 8 7 9 4) with cost=(17.1).

3.5 Select the best mutation (SBM)

This strategy applies various mutation operators simultaneously to the same individual, and from each mutation produces one offspring; the "best" offspring that does not already exist in the population is added to the population [38]. For TSP the "best" solution, is the one with the minimum distance.

In this paper, we used three mutations only (slide mutation, inversion mutation, and the proposed IRGIBNNM), instead using several other mutations as proposed by [38].

A larger example is shown in Figures 5 and 6, which depicts the implementation of four mutations, in addition to the SBM strategy for 80 random cities.

A real data example is shown in Figure 7, which shows the implementation of the four mutations and SBM on a particular route of the TSP (eil51) taken from TSBLIB [45]. A closer look at Figures 5-7 shows significant improvements on the initial tour, particularly when using IRGIBNNM or SBM strategy.

4 Experimental setup and result

To evaluate the performance of the proposed mutation (IRGIBNNM) and the new SBM strategy, we conducted several experiments on twelve TSP instances, each having the known solution (optimal). Those instances were taken from the TSPLIB [45], and they include: eil51, a280, bier127, berlin52, KroA100, KroA200, ch150, rat195, st70, pr125, pr226 and lin318. Same experiments, under the same circumstances were conducted to examine the convergence to a minimum value of each operator separately, including the other mutations (slide, inversion)

We have implemented the new SBM strategy the same as proposed by [38], but using three mutations only (slide, inversion, and the proposed IRGIBNNM), instead of several other mutations, considering the best offspring to be added to the population. To prevent duplication of chromosomes, if the new offspring is found in the population, we consider the lower quality offspring, and if all of the three offspring found in the population the operation (on that chromosome) is canceled.

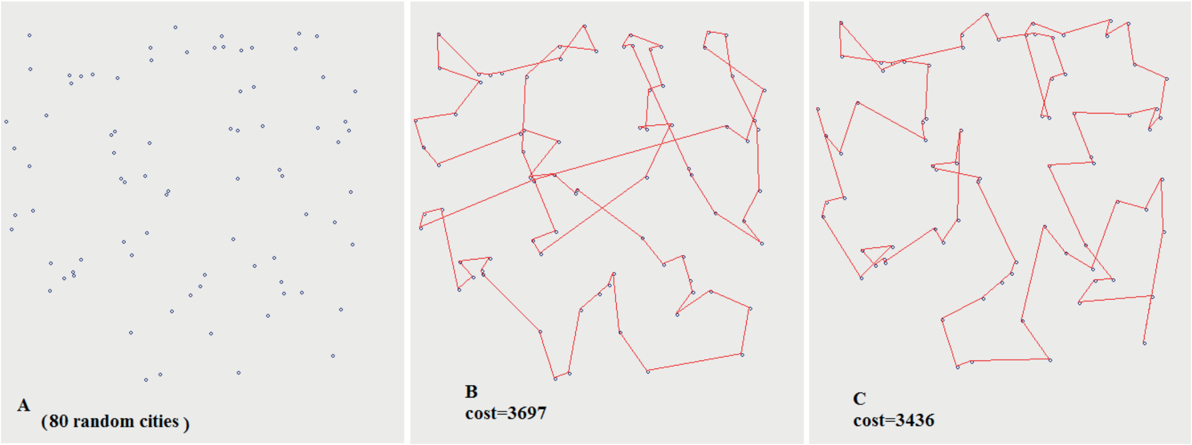

Figure 5 *Implementation of the three mutations for 80 random cities: A) 80 random cities, B) slide mutation, C) inversion mutation*

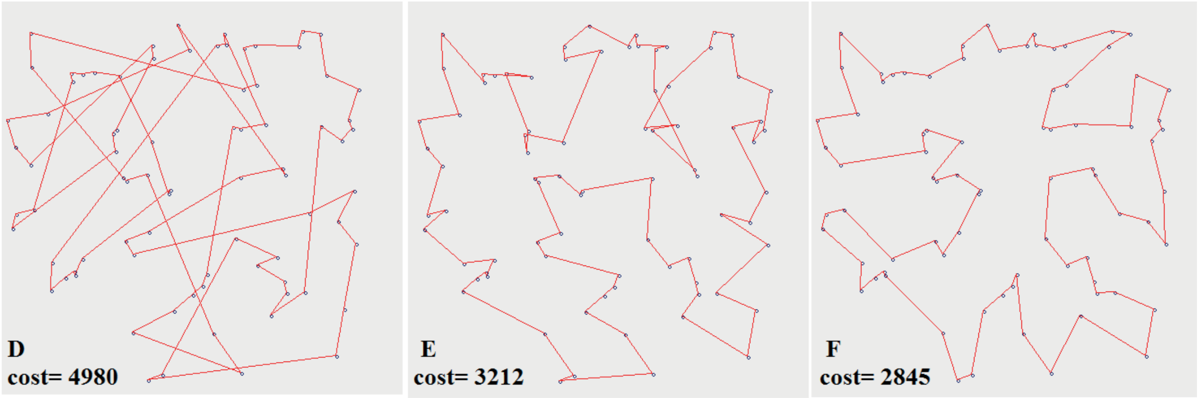

Figure 6 *Implementation of the three mutations on the same random cities of Figure 5: D) RGIBNNM, E) IRGIBNNM, F) SBM strategy*

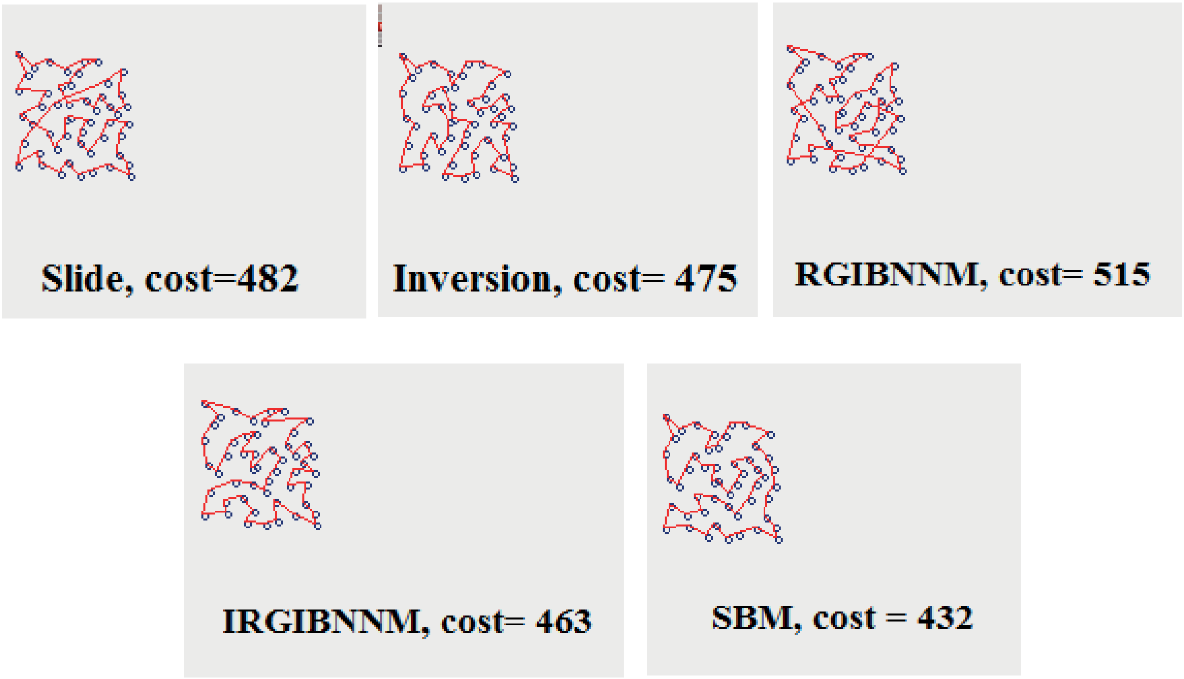

Figure 7 *The effect of applying the mutations on Eil51*

***Table 1** Results of TSP instances obtained by inversion and slide mutations after 2000 generations*

mutation type			inversion mutation			slide mutation	
instances	optimal	best fitness	worst fitness	average fitness	best fitness	worst fitness	average fitness
eil51	426	440	453	446.1	469	583	503.9
a280	2579	9811	10119	9974.2	9532	10522	9917.4
bier127	118282	167565	183857	172867.4	177720	193326	185276.6
kroA100	21282	30310	33413	31925.3	31800	36279	34120.6
berlin52	7542	7769	8515	8038.1	8498	10154	9334.6
kroA200	29368	80906	84555	81958	74586	90348	83529.8
pr125	73682	151643	168468	161445.4	170304	218119	192498
lin318	42029	185852	192611	188931.6	176935	185899	181978.5
pr226	80369	331572	353613	342094.3	345027	377088	360239.9
ch150	6528	13006	13670	13425.1	13129	15221	13778.1
st70	675	758	815	783	787	1004	882.4
rat195	2323	5548	5955	5836.5	5420	6169	5774.8

***Table 2** Results of TSP instances obtained by IRGIBNNM and RGIBNNM mutations after 2000 generations*

mutation type			IRGIBNNM			RGIBNNM	
Instances	optimal	best fitness	worst fitness	average fitness	best fitness	worst fitness	average fitness
eil51	426	448	463	455.3	518	603	575.5
a280	2579	7313	7846	7507.9	6543	8307	7526.5
bier127	118282	156903	169657	164072.9	205820	254541	234760.2
kroA100	21282	25941	29218	27418.7	43474	53903	48077.1
berlin52	7542	8098	8705	8354.2	9639	11105	10296.1
kroA200	29368	59802	63911	62136.9	88409	109892	97125.7
pr125	73682	111055	127783	121013.5	213526	270814	235064.1
lin318	42029	132899	145109	136569.5	159856	178241	173127.6
pr226	80369	191049	234720	216699	288421	380900	322855.1
ch150	6528	10517	11396	11111.9	15071	18435	16774.2
st70	675	733	772	753.4	1058	1296	1222.1
rat195	2323	4321	4758	4554.2	6203	7492	7081.5

In all experiments, our GA used the reinsertion method, which is an expansion sampling [46], where this method means, only the excellent half (from the new individuals and old generation) is selected as a population for the next generation. in other words, when creating a new generation, the old generation competes with the new individuals.

We repeated each experiment 10 times, the GA parameters used are as follows: the Population size=100, the probability of crossover=0% and all previous mutations occur 100%. The initial population is random based population seeding and selection strategy in all algorithms is random. The termination criterion is based on a fixed number of generations reached. For all of our experiments the maximum number of generations=2000.

The operators are coded in VC++, and the computer specifications: 1.66 GHz processor PC with 2 GB of RAM.

The results of the mutations evaluated on 12 instances from the TSP are summarized in Tables 1 and 2.

As can be seen from Tables 1 and 2, the best performance was recorded by the IRGIBNNM for 10 instances, followed by the inversion mutation, which also shows a better performance than both of the slide mutation and the RGIBNNM. The significant performance of the IRGIBNNM is justified by the exploiting of two mutations applied after each other on the same individual. The first provides random solutions and the second provides solutions based on some knowledge of the nearest neighbor. Randomness provided by the inversion mutation, and knowledge provided by

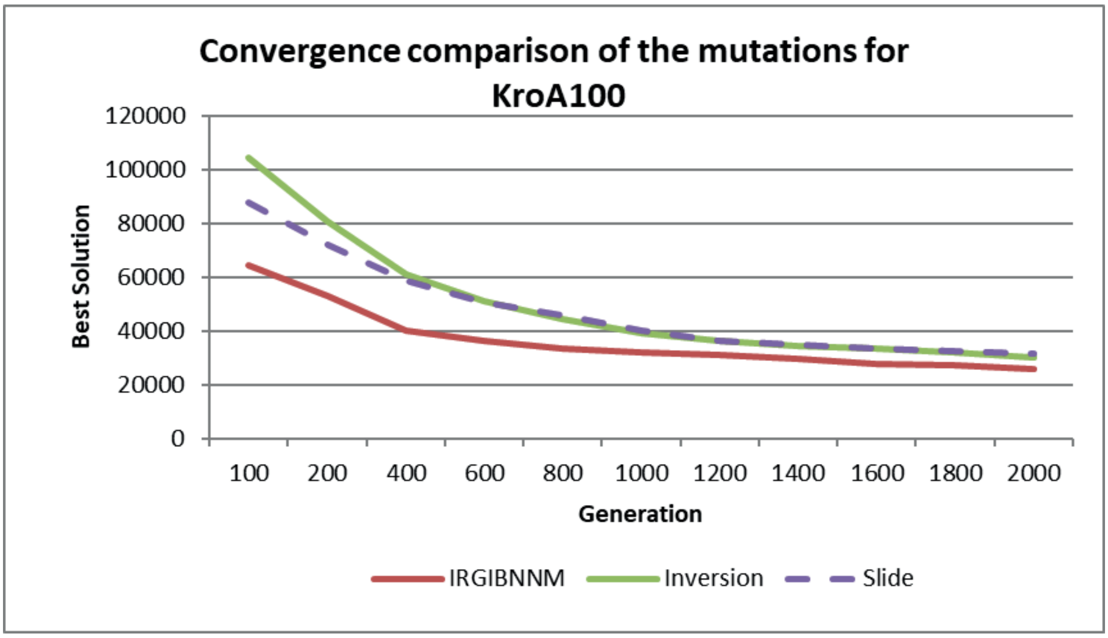

***Figure 8** Mutation's convergence to the minimum solution, kroA100*

***Table 3** Results of TSP instances obtained by SBM after 2000 generations*

Instances	Optimal	Best Fitness	Worst fitness	Average fitness
eil51	426	428	439	432.7
a280	2579	2898	3089	2974.9
bier127	118282	121644	128562	124492.5
kroA100	21282	21344	22788	21957.1
berlin52	7542	7544	8423	7890.7
kroA200	29368	30344	32103	31369
pr152	73682	74777	86240	77022.9
lin318	42029	47006	50033	48234.6
pr226	80369	82579	87006	84409.1
ch150	6528	6737	7044	6876
st70	675	677	723	694.8
rat195	2323	2404	2561	2481.9

the RGIBNNM allow for more diversity of good solutions, which leads to better results.

On result in Table 1 and 2 indicates that the SBM showed faster convergence to the minimum value followed by IRGIBNNM (at the level of mutation alone).

There are several performance factors used to investigate the significance of the importance of the different technique used to improve any GA, such as: computation time, error rate and average convergence [26] .

1- error rate (%): it could be defined as the percentage of difference in the fitness value of the solution with the known optimal solution for the problem. It can be given as:

*error rate (%)=((fitness-optimal fitness)/optimal fitness)*100%.* (1)

2- average convergence (%): it is defined as the average of the convergence rate of solutions.

*average convergence (%)=1-((average fitness - optimal fitness)/optimal fitness)*100.* (2)

Figure 8 shows the convergence to the minimum value recorded by each mutation. Again IRGIBNNM shows faster convergence to the minimum value than the other two mutations on KroA100. This faster convergence is due to the same randomness and knowledge afforded by the IRGIBNNM.

Using the same GA parameters, the second set of experiments is conducted to measure the performance of the new SBM, and to show the effective use of more than one mutation at the same time by the GAs. The results are shown in Table 3 and Figures 9-11.

As can be seen from the results in Table 3, it is important to select the appropriate mutation, in particular for the SBM strategy, and in general for the GA, because the choice of those methods affects the results of the GA

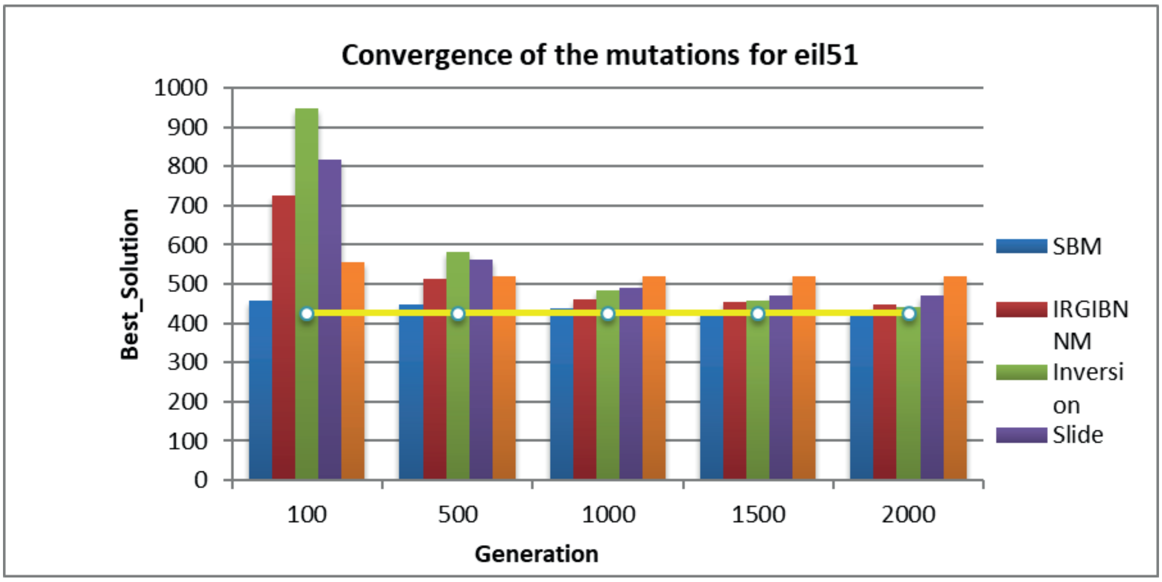

***Figure 9** Convergence Comparison for eil51*

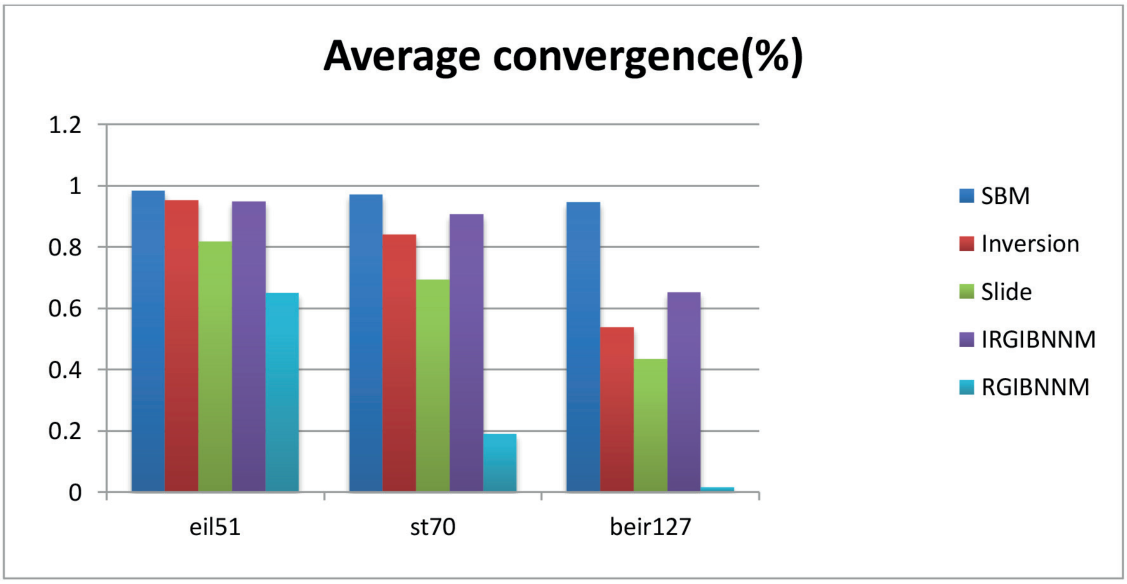

***Figure 10** Average Convergence of 4 mutations and SBM strategy for three instances from TSPLIB (eil51, st70, beir127)*

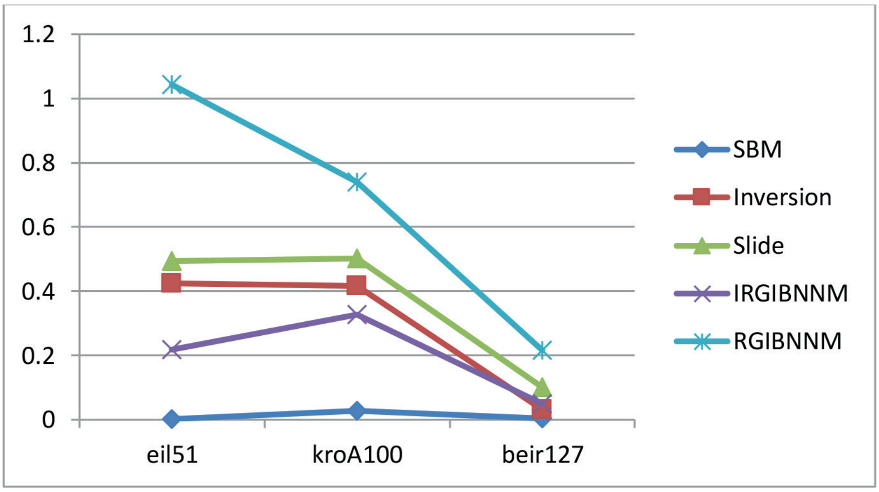

***Figure 11** Error rate of 4 mutations and SBM strategy for three instances from TSPLIB (eil51, st70, beir127)*

***Table 4** Results of new SBM compared to those of old SBM*

Instances	Optimal	New SBM 2000 Generations	Old SBM [38] 2000 Generations
eil51	426	428	443
a280	2579	2898	4824
bier127	118282	121644	175935
kroA100	21282	21344	33739
berlin52	7542	7544	8326
kroA200	29368	30344	51865
pr152	73682	74777	141114
lin318	42029	47006	94865
pr226	80369	82579	207167
ch150	6528	6737	8130
st70	675	677	809
rat195	2323	2404	3790

significantly. The best performance was recorded by the SBM algorithm, followed by the proposed IRGIBNNM, this is seen from Tables 2 and 3.

As seen in Figure 9, the SBM performs better than the other mutations, it is interesting to note that the solutions provided by the SBM are close to the optimal solutions for most of the TSP instances examined. A traditional genetic algorithm commonly uses one mutation operator. We propose using more than one mutation operation, anticipating that different operators will produce different patterns in the offspring and provide some sort of diversity in the population, so as to improve the overall performance of the genetic algorithm [47-48].

Results from Figures 10 and 11 shows the efficient use the three mutations together by the SBM, where the SBM achieves the highest convergence and less error with significant difference.

We justify the significant performance of the SBM as follows, intuitively, we have 2 options for the quality of a solution provided by any mutation, comparing to the average quality in the current population, a) a lower quality solution, and b) a higher quality solution; assuming that a solution with the same quality is considered as a higher quality solution. The new SBM uses 3 mutations, which are applied on the same chromosome, the probability to have them all fail, (i.e. to get lower quality outcomes (offspring) from all mutations used) is 1 out of 8 (low, low, low), while the probability to get a higher quality by any of them is 7 out of 8 possibilities, this high success rate justifies the significant performance of the SBM. Same justification applies to the good performance of the proposed IRGIBNNM, but with a lower success rate of 3 out of 4, since the IRGIBNNM uses only 2 mutations.

Despite the aim of this paper is not to find the optimal solution for TSP, the solutions of the hybrid mutation was close to optimal solutions in some cases, and none could achieve an optimal solution, this indicates the importance of the crossover operator and number of generation and operators ratios and other parameters along with mutation, due to the effective impact of their convergence to an optimal or near optimal solution.

The success of the new SBM is not attributed only to the use of multi mutations as described above, but also to the quality of the solutions provided by the mutations used by the SBM, and this pays attention to the proposed IRGIBNNM, which used by the SBM among the other two mutations. This conclusion is supported by comparing the results of the new SBM with the old SBM proposed by [38], see Table 4 and whith same genetic algorithm parameter, e.g (population size=100, the probability of crossover=0% and all previous mutations occur 100%. The initial population is random based population seeding and selection strategy in all algorithms is random. The termination criterion is based on a fixed number of generations reached. the maximum number of generations=2000.)

Comparing the proposed methods with the plethora of mutations found in the literature is not appreciated, because of the different parameters used by different GAs, such as the number of generations, the mutation rate, crossover rate, population size, selection method, initial population seeding, etc., since each of these parameters affects the results of the GA significantly.

Time complexity for most of mutations found in the literature designed for the TSP ranges from O(1) (such as the simple-random-swapping algorithms) to O(N) (for more complex mutations such as the slide, inversion and RGIBNNM mutations, where N is the number of cities in a TSP instance.

The time complexity of the proposed IRGIBNNM mutation is O(2N), since it uses two mutations of order N, namely, the inversion mutation and the RGIBNNM where each consumes O(N) time.

Accordingly, the Time complexity of the new SBM is O(4N), because it uses 3 mutations, namely slide mutation with O(N), inversion mutation with O(N), and the proposed IRGIBNNM with O(2N), comparing to the old SBM, which has O(10N) as it uses ten O(N) mutations. Asymptotically speaking, both of the proposed methods are of O(N), but in practice they definitely consume more time than most of the mutations found in the literature. Surprisingly, both algorithms might be used to speed up the GA; this is due to their fast convergence to a minimum solution. See Figures 9 and 10, using just the first 100 iterations the GA converged to high quality solutions.

5 Conclusion

In this paper, we propose a hybrid mutation based on knowledge of the TSP and random swapping) called "IRGIBNNM" to enhance the performance of the GA for solving the TSP. We have compared the performance of the IRGIBNNM with three existing mutations, in addition to the SBM, which in this work used three mutations including the proposed one.

The experimental results of 12 TSP instances show the efficiency of the proposed mutation, and the strength of the new SBM, both of the proposed methods benefit from randomness and knowledge provided by the nearest neighbor approach. Also, both methods benefit from the increased probability of getting new high quality solutions due to the use of more than one mutation.

The high quality solutions for the TSP obtained by a GA, which used only the mutation operator, without using other advanced options that used GA by state-of-the-art such as advanced crossovers, initial seeding, advanced selection methods, adaptive change of population size and mutation/crossover rates, etc. The future work will focus on employing the proposed method with other advanced operators to further enhance the performance of the GA when applied for solving the TSP.

References

[1] SONI, N., KUMAR, T. Study of various mutation operators in genetic algorithms. *International Journal of Computer Science and Information Technologies* [online]. 2014, **5**(3), p. 4519-4521. ISSN 0975-9646. Available from: http://ijcsit.com/docs/Volume%205/vol5issue03/ijcsit20140503404.pdf

[2] POTVIN, J.-Y. Genetic algorithms for the traveling salesman problem. *Annals of Operations Research* [online]. 1996, **63**(3), p. 337-370. ISSN 0254-5330, eISSN 1572-9338. Available from: https://doi.org/10.1007/BF02125403

[3] HASSANAT, A. B. A., ALKAFAWEEN, E. On enhancing genetic algorithms using new crossovers. *International Journal of Computer Applications in Technology* [online]. 2017, **55**(3), p. 202-212. ISSN 0952-8091, eISSN 1741-5047. Available from: https://doi.org/0.1504/IJCAT.2017.084774

[4] SAMANTA, S., DE, A., SINGHA, S. Solution of traveling salesman problem on scx based selection with performance analysis using genetic algorithm. *International Journal of Engineering Science and Technology* [online]. 2011, **3**(8), p. 6622-6629. eISSN 0975-5462. Available from: http://www.ijest.info/docs/IJEST11-03-08-258.pdf

[5] RAO, A., HEGDE, S. K. Literature survey on travelling salesman problem using genetic algorithms. *International Journal of Advanced Research in Eduation Technology* [online]. 2015, **2**(1), p. 42-45. ISSN 2394-6814, eISSN 2394-2975. Available from: http://ijaret.com/wp-content/themes/felicity/issues/vol2issue1/anitha.pdf

[6] RAI, K., MADAN, L., ANAND, K. Research paper on travelling salesman problem and it's solution using genetic algorithm. *International Journal of Innovative Research in Technology* [online]. 2014, **1**(11), p. 103-114. ISSN 2349-6002. Available from: http://ijirt.org/master/publishedpaper/IJIRT101672_PAPER.pdf

[7] FREISLEBEN, B., MERZ, P. A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems. In: IEEE International Conference on Evolutionary Computation : proceedings [online]. 1996. ISBN 0-7803-2902-3. Available from: https://doi.org/10.1109/ICEC.1996.542671

[8] LARRANAGA, P., KUIJPERS, C. M. H., MURGA, R. H., INZA, I., DIZDAREVIC, S. Genetic algorithms for the travelling salesman problem: a review of representations and operators. *Artificial Intelligence Review* [online]. 1999, **13**(2), p. 129-170. ISSN 0269-2821, eISSN 1573-7462. Available from: https://doi.org/10.1023/A:1006529012972

[9] DORIGO, M., GAMBARDELLA, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* [online]. 1997, **1**(1), p. 53-66. ISSN 1089-778X, eISSN 1941-0026. Available from: https://doi.org/10.1109/4235.585892

[10] KARKORY, F. A., ABUDALMOLA, A. A. Implementation of heuristics for solving travelling salesman problem using nearest neighbour and minimum spanning tree algorithms. *International Journal of Computer and Information Engineering* [online]. 2013, **7**(10), p. 1524-1534. ISNI 0000000091950263. Available from: https://publications.waset.org/17101/pdf

[11] MALEK, M., GURUSWAMY, M., PANDYA, M., OWENS, H. Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. *Annals of Operations Research* [online]. 1989, **21**(1), p. 59-84. ISSN 0254-5330, eISSN 1572-9338. Available from: https://doi.org/10.1007/BF02022093

[12] GENDREAU, M., HERTZ, A., LAPORTE, G. A tabu search heuristic for the vehicle routing problem. *Management Science* [online]. 1994, **40**(10), p. 1276-1290. ISSN 0025-1909, eISSN 1526-5501. Available from: https://doi.org/10.1287/mnsc.40.10.1276

[13] SHI, X. H., LIANG, Y. C., LEE, H. P., LU, C., WANG, Q. X. Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters* [online]. 2007, **103**(5), p. 169-176. ISSN 0020-0190, eISSN 1872-6119. Available from: https://doi.org/10.1016/j.ipl.2007.03.010

[14] DURBIN, R., SZELISKI, R., YUILLE, A. An analysis of the elastic net approach to the traveling salesman problem. *Neural Computation* [online]. 1989, **1**(3), p. 348-358. ISSN 0899-7667, eISSN 1530-888X. Available from: https://doi.org/10.1162/neco.1989.1.3.348

[15] AARTS, E. H., STEHOUWER, H. P. Neural networks and the travelling salesman problem. In: ICANN'93 : proceedings. 1993.

[16] ALKAFAWEEN, E. *Novel methods for enhancing the performance of genetic algorithms*. Master thesis. Jordan: Mutah University, 2015.

[17] GOLDBERG, D. E. *Genetic algorithms in search, optimization, and machine learning*. 1.ed. Boston: Addison-Wesley Professional, 1989. ISBN 978-0201157673.

[18] HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, MA: MIT Press, 1975. ISBN 9780262082136.

[19] KOREJO, I., YANG, S., LI, CH. A directed mutation operator for real coded genetic algorithms. In: European Conference on the Applications of Evolutionary Computation 2010 : proceedings [online]. Berlin, Heidelberg: Springer, 2010. ISBN 978-3-642-12238-5, eISBN 978-3-642-12239-2. p. 491-500. Available from: https://doi.org/10.1007/978-3-642-12239-2_51

[20] MOHAMMED, A. A., NAGIB, G. Optimal routing in ad-hoc network using genetic algorithm. *International Journal of Advanced Networking and Applications* [online]. 2012, **3**(05), p. 1323-1328. eISSN 0975-0282. Available from: https://www.ijana.in/papers/V3I5-4.pdf

[21] BENKHELLAT, Z., BELMEHDI, A. Genetic algorithms in speech recognition systems. In: 2012 International Conference on Industrial Engineering and Operations Management : proceedings. 2012. ISBN 9781629939117, p. 853-858.

[22] PAULINAS, M., USINSKAS, A. A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and Control*. 2015, **36**(3), p. 278-284. ISSN 1392-124X, eISSN 2335-884X.

[23] SRIVASTAVA, P. R., KIM, T.-H. Application of genetic algorithm in software testing. *International Journal of Software Engineering and its Applications*. 2009, **3**(4), p. 87-96. ISSN 1738-9984.

[24] MICHALEWICZ, Z. *Genetic algorithms+data structures=evolution programs* [online]. 3. ed. Berlin Heidelberg: Springer Science & Business Media, 2013. ISBN 978-3-540-60676-5, eISBN 978-3-662-03315-9. Available from: https://doi.org/10.1007/978-3-662-03315-9

[25] YUGAY, O., KIM, I., KIM, B., KO, F. I. Hybrid genetic algorithm for solving traveling salesman problem with sorted population. In: 2008 Third International Conference on Convergence and Hybrid Information Technology : proceedings [online]. 2008. ISBN 978-0-7695-3407-7. Available from: https://doi.org/10.1109/ICCIT.2008.373

[26] HASSANAT, A., PRASATH, V., ABBADI, M., ABU-QDARI, S., FARIS, H. An improved genetic algorithm with a new initialization mechanism based on regression techniques. *Information* [online]. 2018, **9**(7), p. 167-196. eISSN 2078-2489. Available from: https://doi.org/10.3390/info9070167

[27] KOREJO, I. A., YANG, S., BROHI, K., KHUHRO, Z. U. A. Multi-population methods with adaptive mutation for multi-modal optimization problems. *International Journal on Soft Computing, Artificial Intelligence and Applications* [online]. 2013, **2**(2), p. 1-11. ISSN 2319-4081, eISSN 2319-1015. Available from: https://doi.org/10.5121/ijscai.2013.2201

[28] FRIEDRICHS, T., OLIVETO, P. S., SUDHOL, D., WITT, C. Analysis of diversity-preserving mechanisms for global exploration. *Evolutionary Computation* [online]. 2009, **17**(4), p. 455-476. ISSN 1063-6560, eISSN 1530-9304. Available from: https://doi.org/10.1162/evco.2009.17.4.17401

[29] YANG, S. Adaptive non-uniform mutation based on statistics for genetic algorithms. In: 4th Annual Conference on Genetic and Evolutionary Computation GECCO'02 : proceedings. Part II. 2002. ISBN 978-1-55860-878-8, p. 650-657.

[30] HASSANAT, A., ALMOHAMMADI, K., ALKAFAWEEN, E. ABUNAWAS, E., HAMMOURI, A., PRASATH, V. B. S. Choosing mutation and crossover ratios for genetic algorithms - a review with a new dynamic approach. *Information* [online]. 2019, **10**(12), p. 390-426. eISSN 2078-2489. Available from: https://doi.org/10.3390/info10120390

[31] NICOARA, E. S. Mechanisms to avoid the premature convergence of genetic algorithms. *Petroleum - Gas University of Ploiesti Bulletin - Mathematics, Informatics, Physics Series* [online]. 2009, **61**(1), p. 87-96. ISSN 1224-4899, eISSN 2067-242X. Available from: http://bulletin-mif.unde.ro/docs/20091/12NICOARA_SIMONA.pdf

[32] ABDOUN, O., ABOUCHABAKA, J. A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. *International Journal of Computer Applications* [online]. 2011, **31**(11), p. 49-57. ISSN 0975-8887. Available from: https://arxiv.org/ftp/arxiv/papers/1203/1203.3097.pdf

[33] SIVANANDAM, S. N., DEEPA, S. N. *Introduction to genetic algorithms* [online]. 1. ed. Berlin Heidelberg: Springer -Verlag, 2008. ISBN 978-3-540-73189-4, eISBN 978-3-540-73190-0. Available from: https://doi.org/10.1007/978-3-540-73190-0

[34] BANZHAF, W. The "molecular" traveling salesman. *Biological Cybernetics* [online]. 1990, **64**(1), p. 7-14. ISSN 0340-1200, eISSN 1432-0770. Available from: https://doi.org/10.1007/BF00203625

[35] MICHALEWICZ, Z. *Genetic algorithms+data structures=evolutionary programs* [online]. 1. ed. Berlin Heidelberg: Springer-Verlag, 1992. ISSN 1431-0066, eISBN 978-3-662-02830-8. Available from: https://doi.org/10.1007/978-3-662-02830-8

[36] FOGEL, D. A. A parallel processing approach to a multiple travelling salesman problem using evolutionary programming. In: Fourth annual Symposium on Parallel Processing : proceedings. IEEE Orange County Computer Society, 1990.

[37] LOUIS, S. J., TANG, R. Interactive genetic algorithms for the traveling salesman problem. In: GECCO-99: Genetic and Evolutionary Computation Conference : a Joint Meeting of the Eighth International Conference on Genetic Algorithms (ICGA-99) and the Fourth Annual Genetic Programming Conference (GP-99) : proceedings. Vol. 1. Morgan Kaufmann Publishers, 1999. ISBN 1558606114, 9781558606111.

[38] HASSANAT , A. A. B., ALKAFAWEEN, E., AL-NAWAISEH, N. A., ABBADI, M. A., ALKASASSBEH, M., ALHASANAT, M. B. Enhancing genetic algorithms using multi mutations: experimental results on the travelling salesman problem. *International Journal of Computer Science and Information Security* [online]. 2016, **14**(7), p. 785-801. ISSN 1947-5500. Available from: https://arxiv.org/ftp/arxiv/papers/1602/1602.08313.pdf

[39] HONG, T. P., WANG, H. S., LIN, W. Y., LEE, W. Y. Evolution of appropriate crossover and mutation operators in a genetic process. *Applied Intelligence* [online]. 2002, **16**(1), p. 7-17. ISSN 0924-669X, eISSN 1573-7497. Available from: https://doi.org/10.1023/A:1012815625611

[40] DENG , Y. LIU, Y., ZHOU, D. An improved genetic algorithm with initial population strategy for symmetric TSP. *Mathematical Problems in Engineering* [online]. 2015, Article ID 212794. ISSN 1024-123X, eISSN 1563-5147. Available from: https://doi.org/10.1155/2015/212794

[41] HILDING, F., WARD, K. Automated crossover and mutation operator selection on genetic algorithms. In: 9th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems : proceedings [online]. Berlin Heidelberg: Springer-Verlag, 2005. ISBN 978-3-540-28897-8, eISBN 978-3-540-31997-9. Available from: https://doi.org/10.1007/11554028

[42] KATAYAMA, K., SAKAMOTO, H., NARIHISA, H. The efficiency of hybrid mutation genetic algorithm for the travelling salesman problem. *Mathematical and Computer Modelling*. 2000, **31**(10-12), p. 197-203. ISSN 0895-7177.

[43] HONG, T. P., WANG, H. S., CHEN, W. CH. Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of Heuristics* [online]. 2000, **6**(4), p. 439-455. ISSN 1381-1231, eISSN 1572-9397. Available from: https://doi.org/10.1023/A:1009642825198

[44] SUN, W. A novel genetic admission control for real-time multiprocessor systems. In: 2009 International Conference on Parallel and Distributed Computing, Applications and Technologies : proceedings [online]. 2009. ISSN 2379-5352, eISBN 978-0-7695-3914-0. Available from: https://doi.org/10.1109/PDCAT.2009.10

[45] REINELT, G. TSPLIB [online]. University of Heidelberg, 1996. Available from: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95

[46] DONG, M., WU, Y. Dynamic crossover and mutation genetic algorithm based on expansion sampling. In: International Conference on Artificial Intelligence and Computational Intelligence AICI 2009 : proceedings [online]. Lecture Notes in Computer Science, vol. 5855. Berlin, Heidelberg: Springer, 2009. ISBN 978-3-642-05252-1, eISBN 978-3-642-05253-8. Available from: https://doi.org/10.1007/978-3-642-05253-8_16

[47] CONTRERAS-BOLTON, C., PARADA, V. Automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. *PlOS one* [online]. 2015, **10**(9). eISSN 1932-6203. Available from: https://doi.org/10.1371/journal.pone.0137724

[48] SPEARS, W. M. Adapting crossover in evolutionary algorithms. In: *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*. MCDONNELL, J. R., REYNOLDS, R. G., FOGEL, D. B. (eds.). 1995. eISBN 9780262290920. Available from: https://doi.org/10.7551/mitpress/2887.001.0001