

# Reducing Premature Convergence Problem in Genetic Algorithm: Application on Travel Salesman Problem

Saleem Zeyad Ramadan<sup>1</sup>

<sup>1</sup> Mechanical and Industrial Engineering Department, Applied Science University, Amman, Jordan

Correspondence: Saleem Zeyad Ramadan, Mechanical and Industrial Engineering Department, Applied Science University, P.O. Box 166, Amman 11931, Jordan. E-mail: s\_ramadan@asu.edu.jo

Received: June 27, 2012 Accepted: December 13, 2012 Online Published: December 24, 2012

doi:10.5539/cis.v6n1p47

URL: <http://dx.doi.org/10.5539/cis.v6n1p47>

*The author is grateful to the Applied Science Private University, Amman, Jordan, for the financial support granted to this research (Grant No. DRGS-2011-34)*

## Abstract

Genetic algorithm (GA) is based on Darwin's natural selection theory and is used extensively in combinatorial problems as these problems are demanding in terms of computational time. GA shows very good results in terms of both computational time and quality of solution for combinatorial problems as GAs have some traits that make them one of the best evolutionary algorithms (EAs). The use of both mutation and crossover operators make them, relative to other EAs, highly immune to be trapped in a local optima and thus less vulnerable to premature convergence problem. Traditionally, the solution for premature convergence problem is to maintain a certain degree of diversity of the GA's population without affecting the convergence process itself. In this paper, this concept has been practiced where Frequency Crossover strategy (FC) along with nine different mutation strategies have been proposed and applied to travel salesman problem (TSP) to reduce the effect of premature convergence problem. Three sets of benchmark data have been used to test the effectiveness of this GA. The results showed that both the nine mutation types and the FC are essential for the proposed GA to perform well. While this GA has been applied on TSP in this paper, it is also believed that it is applicable on any problem that has an Order-Based chromosome representation.

**Keywords:** evolutionary algorithms, genetic algorithms, premature convergence problem, travel salesman problem

## 1. Introduction

Genetic algorithm is relatively new method in the optimization field and is not widely used even though it is fast and powerful in finding an optimal or near optimal solutions. This technique was first introduced in the seventies based on the well-known theory of Darwin's natural selection. Darwin developed his theory based on the natural selection phenomena which can be summarized as follows: individuals that fit better to their surrounding have a greater probability to survive in the process of struggle for existence. The essence of Darwin's theory is that evolution occurs if three necessary and sufficient conditions are met. These conditions are: struggle for existence, variation, and inheritance of the variation traits among the individuals of the populations. These conditions are necessary means that if for any reason one of these conditions does not happen in any generation, natural selection (and thus evolution) will cease at that generation. Therefore, in genetic algorithm premature convergence problem may happen if the population's individuals at any generation do not show variation in their traits and/or if the traits that cause variation among the individuals in the population do not have a heritable basis. To say that the three conditions are sufficient means that if all three conditions are met, natural selection will unavoidably happen and this can lead to a change in the characteristics of the population from one generation to the next, i.e., evolution definitely occurs.

When designing GA, the designers should keep in mind these three conditions to ensure that the evolution process will persist and to avoid premature convergence in their GA. On one hand, Using crossover and mutation operators properly help in creating variation among the individuals in the population thus helping in reducing the premature convergence problem (Qu, 2002). On the other hand, if variations among the individuals in the

population are very high, the information that is accumulated and carried over the generations in the individuals about the solution of the problem will be lost and thus mature convergence can be very hard to achieve. Therefore, a good design of a GA should be able to compromise between diversity and stability of the individuals in the population.

The population diversity plays a major role in the performance of GA (Popovic, 1997). It is widely agreed among GA developers that the higher the diversity in the population the less premature convergence chance to happen and thus the higher chance to escape from a local optima. Different crossover strategies were proposed in the literature to create diversity in the population. Goldberg and Lingle proposed the Partially-Mapped Crossover Operator (PMX) where a segment of one parent's chromosome is mapped into a segment of another parent's chromosome and the remaining genes are exchanged (Goldberg & Lingle, 1985). Oliver et al. proposed the Cycle Crossover operator (CX). This operator creates the offspring from the parents by copying the value of the gene along with its position from the parents into the offspring taking into consideration the feasibility of the chromosome (Oliver, Smith, & Holland, 1987). Brady proposed Sorted Match Crossover strategy in which similar sub tours between the parents are first determined and then the offspring is generated from the parents with the highest cost where this sub tour is replaced by the lowest sub tour's cost (Brady, 1985).

Ursem tried to solve this problem by having a balance between crossover and mutation rates under a certain threshold values (Ursem, 2002). Others tried to minimize the effect of this problem by manipulating the chromosome representation using odd and even representation (Abu Bakar, Ramlan, & Muhammad, 2007). Social Disaster Technique (SDT) was proposed to avoid the premature convergence to local optima in the TSP. The authors introduced two operators, Packing and Judgment Day, to achieve this goal (Kureichick et al., 1996). A comprehensive review on representations and operators for TSP can be found in reference (Larra et al., 1999).

In this paper, Frequency Crossover (FC) along with nine different types of mutations will be proposed to solve the TSP. The FC will be used to stabilize the population while the nine different types of mutations will be used to increase the diversity of the population to prevent premature convergence problem. The algorithm will be tested using standard problems eil51, eil76 and eil101 found in the TSPLIB95 (Reinelt, 1995).

The rest of the paper will be organized as follows: A description of TSP will be presented, followed by a full description of the proposed GA, and finally, the results and conclusions obtained from the proposed GA will be discussed.

## 2 Travel Salesman Problem

Travel salesman problem is a classical problem that is known to be an NP-hard problem in Combinatorial Optimization. The problem can be summarized as follows: given a list of  $n$  cities and the distance (cost) matrix,  $d_{ij}$ , between each pair of cities, the problem is to find the shortest (cheapest) round trip that involves all of the  $n$  cities in it and in the same time ensures that each city will be visited only once. Traditionally, this problem was solved using integer programming (IP) and can be formulated mathematically as:

$$\text{Minimize : } \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (1)$$

$$\sum_{j=1}^n x_{ij} = 1, \forall i \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \forall j \quad (3)$$

$$|S| = n \quad (4)$$

$$x_{ij} \text{ are integers} \quad (5)$$

The indices  $i$  and  $j$  represent the city number and can take any value from  $1$  to  $n$ . Equation (1) calculates the total distance (cost) for the round trip where  $x_{ij}$  is a binary variable that takes  $0$  if no direct trip was made between city  $i$  and city  $j$  and takes  $1$  otherwise. Equation (2) and Equation (3) guarantee that each city will be visited only once. Equation (4) forces the cardinality of the cities set in any solution  $S$  to be the number of cities  $n$  and hence will eliminate any sub tour solution.

### 3. The Genetic Algorithm

#### 3.1 Chromosome Representation

The chromosome representation (shown in Figure 1) used in this GA consists of  $n$  distinct integer genes, therefore, each chromosome will represent a feasible tour.

1	2	3	.	.	.	.	.	.	.	.	.	.	.	.	.	$n-2$	$n-1$	$n$
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-------	-------	-----

Figure 1. Chromosome representation for TSP

The phenotype can be easily retrieved from the genotype in this representation as each gene in the genotype represents a city. Cities will be assigned a sequential numbers from 1 to  $n$  and the sequence of the genes will represent the sequence at which the cities will be visited in the round trip. This chromosome representation is known in literature as Order-Based Representation (OBR). In this chromosome representation, the gene carries two pieces of information, the city name, which is represented by the value of the gene, and the sequence of the city in the tour, which is the same as the sequence of the gene in the chromosome. This chromosome representation demands special recombination strategies that take into consideration the special characteristics of this representation. These strategies must ensure the feasibility of the offspring to avoid repair actions to retrieve feasibility for the infeasible offspring.

#### 3.2 Fitness Function

Using OBR permits an easy calculation for the fitness value of the chromosome since the fitness value can be calculated easily by adding the distances between each two adjacent genes (cities) in the tour and then adding the distance between the last gene and the first gene in the tour to accommodate for the round trip path distant.

#### 3.3 Initial Population

The initial population,  $N$  chromosomes, will be generated randomly and the corresponding fitness values of these chromosomes will be calculated.

#### 3.4 Frequency Crossover

$Y$  different chromosomes will be selected randomly, at a time, from the existing population. Chromosomes will be sorted in an ascending order so that the best half of the chromosomes, i.e. first  $Y/2$  chromosomes, will form a crossover group for crossover operator with a 100% crossover rate. The Frequency Crossover strategy proposed in this GA is as follows:

The best chromosome in the crossover group will crossover with every other chromosome in the group to produce one offspring for each crossover. This means that this crossover strategy will produce  $Y/2-1$  offspring. The frequency for the values of the corresponding genes in both parents will be evaluated. Frequency can be either one or two. A frequency of two means that both corresponding genes carry the same city while a frequency of one means that the two corresponding genes have different cities. The genes with frequency of two will be determined and their values (cities numbers) will be copied into the offspring while preserving their relative positions. The complimentary set of cities, the values of the genes that have a frequency equal to one, will be assigned to the remaining genes in the offspring randomly. Figure 2 shows an example for this type of crossover.

Parent 1																	
17	1	12	14	3	8	9	2	4	7	5	10	6	11	15	18	13	16
Parent 2																	
17	9	18	14	3	10	4	1	12	7	5	2	11	6	15	8	13	16
Offspring																	
17	10	1	14	3	11	2	12	9	7	5	4	18	8	15	6	13	16

Figure 2. Frequency crossover strategy

From Figure 2, the cities 17, 14, 3, 7, 5, 15, 13, and 16 have the same locations in both parents and hence the frequency for the genes carrying these cities is 2. These genes i.e., genes number 1, 4, 5, 10, 11, 15, 17, and 18 will be copied to the offspring with their values. The remaining cities (1, 2, 4, 6, 8, 9, 10, 11, 12, 18) will be assigned randomly to the remaining genes (2, 3, 6, 7, 8, 9, 10, 12, 13, 14, 16) in the offspring. This will

guarantee that the generated offspring will be feasible. As generations go on, the order of the cities in the round trip will be stabilized. This will help in preserving the information carried and accumulated over generations about the optimal round trip. Of course, this will reduce the diversity of the population and thus will increase the chances of having premature convergence in the population. Therefore, the FC can be considered as a stability inducer in this GA.

The selected parents and the generated offspring will replace the existing  $Y-1$  chromosomes. The remaining chromosome (since we have  $Y/2$  parents and  $Y/2-1$  offspring) will be immigrated.

The new  $Y$  chromosomes will form a sub population that will be used in the mutation step. The best chromosome in this sub population will be subjected to nine different types of mutations. The best chromosome along with the nine offspring generated by mutation will replace the original sub population. This will increase the diverse of the population and in the same time will guarantee that the best chromosome found in the existing generation will survive to the next generation. These nine mutation strategies can be considered as a diversity inducer in this GA.

### 3.5 Mutation

Nine different types of mutations are proposed and will be grouped into two groups:

Group1 will include the mutation types that will change more than two links in the round trip and Group 2 will include the mutation types that will change only two links in the round trip. Therefore, Group 1 mutations will be used for exploration purposes, since they will cause big change in the chromosomes, while Group 2 mutations will be used for exploitation purposes since they will cause a minor change in the chromosomes. The two groups together will guarantee a high degree of diversity in the population and thus will help in reducing the premature convergence problem in this GA.

It is hoped that the net effect of using the FC, which will stabilize the population, and the nine mutation types, which will increase the diversity of the population, is that the diversity of the population will be maintained and in the same time the mature convergence of the algorithm will not be halted.

It is worth to mention here that there is a relation between the number of mutation types and the size of the sub population  $Y$  such that  $Y$  is always bigger than the number of mutation types by one, i.e.  $Y$  is 10. This will impose a constraint on the population size ( $N$ ) used in this GA such that the population size should be divisible by 10.

#### 3.5.1 Group 1 Mutations

This group of mutations consists of five different types of mutations. This group will be used for exploration purposes since three or four links, depending on the mutation type, will be affected in the round trip. This will produce a relatively big change in the chromosome structure and thus in the fitness value of the chromosome. Therefore, it is hoped that these mutations will explore the solution space adequately.

1) Ends Exchange mutation (EsEm): One gene at position ( $C1$ ) will be chosen at random and the genes between the first gene and  $C1$  will be exchanged with the last (same number) genes. In this mutation, the position of the cutting point ( $C1$ ) must be less than or equal to  $n/2$  to ensure the feasibility of the offspring. Figure 3 shows this mutation.

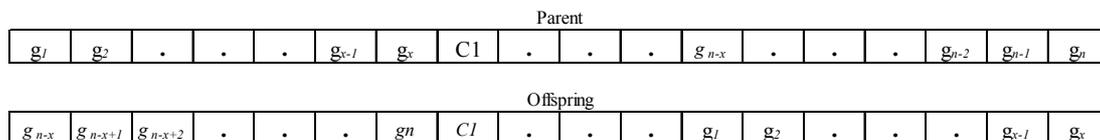


Figure 3. Schematic diagram for the ends exchange mutation

There are three links that were affected by this mutation namely: the link  $g_x-c_1$  changed to  $g_n-c_1$ , the link  $g_{n-x-1}-g_{n-x}$  changed to  $g_{n-x-1}-g_1$ , and the link  $g_n-g_1$  changed to  $g_x-g_{n-x}$ .

2) Group Insertion mutation (GIm): One gene at position ( $C1$ ) will be chosen at random and the genes between the first gene and  $C1$ , which form a set of length  $L$ , will be inserted after  $L$  genes. In this mutation, the position of the cutting point ( $C1$ ) must be less than or equal to  $n/2$  to ensure the feasibility of the offspring. Figure 4 shows this mutation.

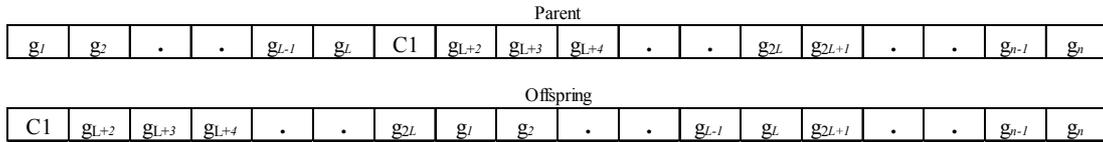


Figure 4. Schematic diagram for the group insertion mutation

There are three links that were affected by this mutation namely: the link  $c_1-g_{L+2}$  changed to  $g_{2L}-g_1$ , the link  $g_{2L}-g_{2L+1}$  changed to  $g_L-g_{2L+1}$ , and the link  $g_n-g_1$  changed to  $g_n-c_1$ .

Another mutation of this kind will be made and will be called Group Insertion mutation 2 (GI2). In this strategy, the last L genes will be inserted after gene number  $n-2L$ . Again, three links will be affected.

3) Reverse Ends mutation (REsm): Two genes at positions (C1 and C2) will be chosen at random and the order of the genes between positions 1 to C1 and between positions C2 to n will be reversed. Figure 5 shows a schematic diagram for this mutation.

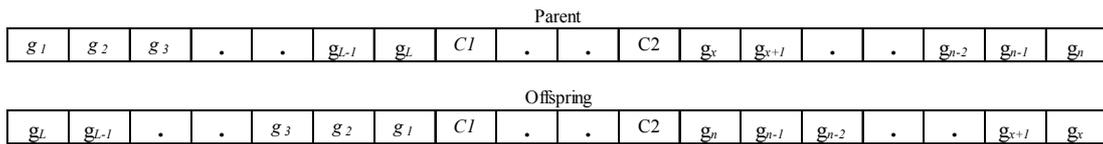


Figure 5. Schematic diagram for reverse ends mutation

There are three links that were affected by this mutation namely: the link  $g_n-g_1$  changed to  $g_x-g_L$ , the link  $c_2-g_x$  changed to  $c_2-g_n$ , and the link  $g_L-c_1$  changed to  $g_1-c_1$ .

4) Two Genes Exchange mutation (TGsEm): Two genes at positions (C1 and C2) will be chosen at random and will be exchanged. Figure 6 shows a schematic diagram for this mutation.

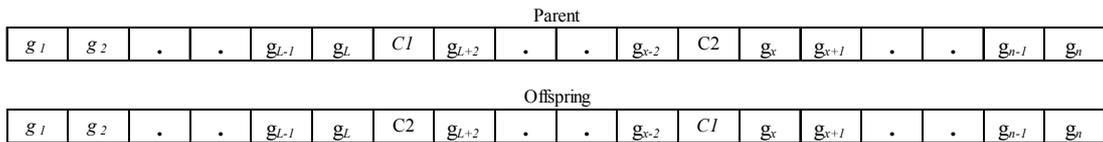


Figure 6. Schematic diagram for two genes exchange mutation

There are four links that were affected by this mutation namely: the link  $g_L-c_1$  changed to  $g_L-c_2$ , the link  $c_1-g_{L+2}$  changed to  $c_2-g_{L+2}$ , the link  $g_{x-2}-c_2$  changed to  $g_{x-2}-c_1$ , the link  $c_2-g_x$  changed to  $c_1-g_x$ .

### 3.5.2 Group 2 Mutations

This group consists of four different types of mutations. It will be used for exploitation purposes because only two links in the round trip will be affected by each type of the mutation.

1) Reverse Ends Exchange mutation (REsEm): One gene at position (C1) will be chosen at random and the genes between the first gene and C1 will be reversed and exchanged with the last (same number) genes after reversing them. In this mutation, the position of the cutting point (C1) must be less than or equal to  $n/2$  to ensure the feasibility of the offspring. Figure 7 shows this mutation.

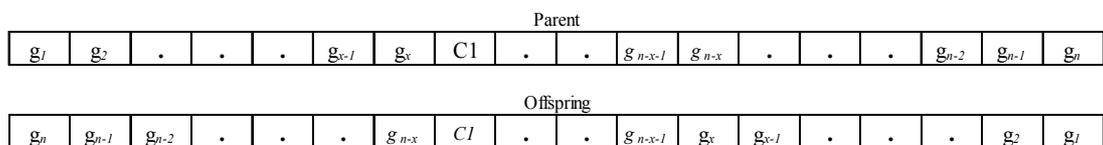


Figure 7. Schematic diagram for reverse end exchange mutation

There are two links only that were affected by this mutation namely: the link  $g_x-c_1$  changed to  $g_{n-x}-c_1$  and the link  $g_{n-x-1}-g_{n-x}$  changed to  $g_{n-x-1}-g_x$ . Since the TSP under consideration is a symmetric TSP, the link  $g_n-g_1$  is the same as the link  $g_1-g_n$ .

2) Reverse End mutation (REm): One gene at position (C1) will be chosen at random and the genes between positions 1 and C1 will be reversed. Figure 8 shows this mutation.

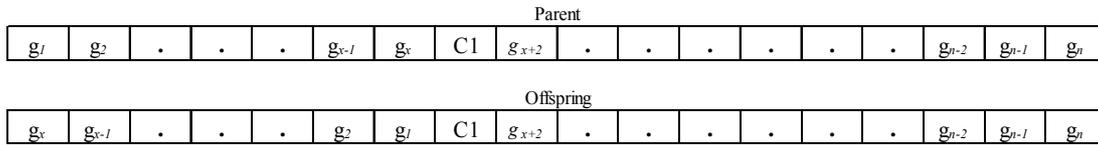


Figure 8. Schematic diagram for reverse end mutation

There are two links only that were affected by this mutation namely: the link  $g_x-c_1$  changed to  $g_1-c_1$  and the link  $g_n-g_1$  changed to  $g_n-g_x$ .

3) One Position Swap mutation (OPSm): One gene at position (C1) will be chosen at random and it will be replaced with the next gene. In this mutation, the position of the cutting point (C1) must be less than or equal to  $n-1$  to ensure the feasibility of the offspring. Figure 9 shows this mutation.

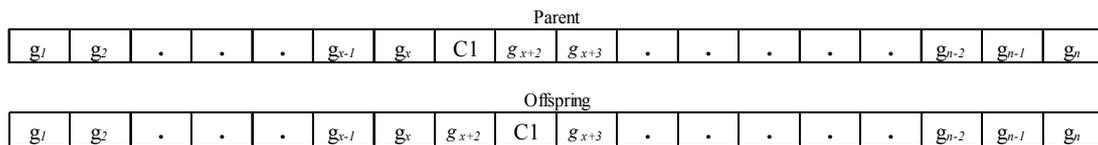


Figure 9. Schematic diagram for one position swap mutation

There are two links only that were affected by this mutation namely: the link  $g_x-c_1$  changed to  $g_x-g_{x+2}$  and the link  $g_{x+2}-g_{x+3}$  changed to  $c_1-g_{x+3}$ . Since the TSP under consideration is a symmetric TSP, the link  $c_1-g_{x+2}$  is the same as the link  $g_{x+2}-c_1$ .

4) Middle Reverse mutation (MRm): Two genes with positions (C1 and C2) will be chosen at random and the order of the genes between them will be reversed. Figure 10 shows a schematic diagram for this mutation.

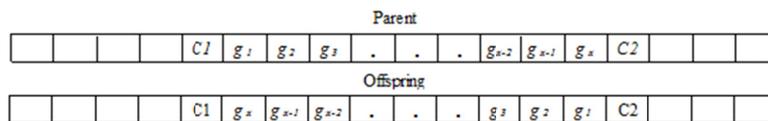


Figure 10. Schematic diagram for middle reverse mutation

There are two links that were affected by this mutation namely: the link  $c_1-g_1$  changed to  $c_1-g_x$  and the link  $g_x-c_2$  changed to  $g_1-c_2$ .

### 3.6 The Termination Criterion

The GA will terminated when 1000 generations are reached.

The pseudo code for the proposed GA can be outlined as follows:

BEGIN GA

Randomly generate N chromosomes

Evaluate the fitness value for each chromosome

Assign the lowest fitness value for BestFitnessValue variable

Assign the route with the lowest fitness value to BestRouteFound

Do while maximum generation number is not reached

    Do until all chromosomes in the existing population have been chosen

        Select ten new chromosomes (not selected before) from the existing population

```

    Evaluate the fitness value for each chromosome
    Select the best chromosome from these ten chromosomes
    Do Frequency Crossover to get ten chromosomes
    Evaluate the fitness value for each one of the ten chromosomes
    Do Group 1 and Group 2 mutations on the best chromosome to generate nine offspring and keep
    the best chromosome
    If the fitness value for any one of the nine offspring is better than the BestFitnessValue
        Update the value of the BestFitnessValue.
        Update the value of the BestRouteFound
    End if
    Update the existing population.
End until loop
End While loop
Present the best solution found by presenting the BestRouteFound and BestFitnessValue
End GA

```

#### 4. Results and Discussion

The proposed GA will be tested using three different sets of data for TSP taken from TSPLIB95 [Rei95]. Table 1 lists the problems along with their relevant parameters.

Table 1. TSPs used for benchmark

Problem	Number of nodes	Best Solution Ever Found
Eil51	51	426
Eil76	76	538
Eil101	101	629

The data given in these instances is an ordered pairs of  $x$  and  $y$  coordinates for the cities locations. Of course, from this  $x$  and  $y$  coordinates, the distant matrix for the different symmetric TSPs can be calculated.

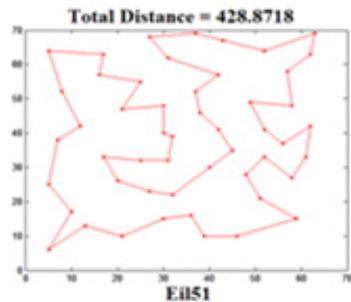
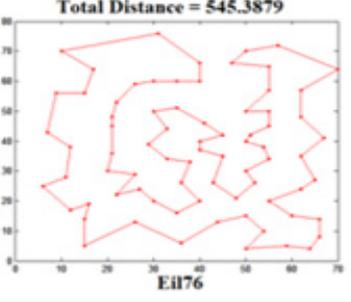
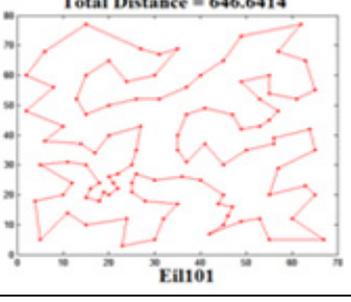
Applying the proposed GA, the results that represent the average of the minimum distant for 20 replications can be summarized in Table 2 for all of the three problems (Eil51, Eil76, and Eil101). The machine used to solve the problems has the following specifications: Manufacturer HP, Model HPE-500f, Processor AMD phenon (tn) IIX6 1045T processor 2.70GHz, RAM 8.0 GB, system 64-bit operating system.

Table 2. Experimental results for Eil51, Eil76, and Eil101

TSP	Average best distant for the proposed GA	Standard deviation	Best found in all replications	Frequency of best found	Average running time
Eil51	429.29	0.8815	428.87	15	352 S
Eil76	546.02	1.2768	545.37	12	451 S
Eil101	648.22	1.4532	646.64	8	590 S

For sack of completion, Table 3 lists the best routes found for each problem that corresponds to the lowest distant found in the 20 replications along with the actual routes.

Table 3. List of best routes found for each TSP

TS Problem	Best Route found	
Eil51	18-4-17-37-5-38-11-32-1-22-8-26-31-28-3-36-35-20-29-2-16-50-21-34-30-9-49-10-39-33-45-15-44-42-19-40-41-13-25-14-24-43-7-23-48-6-18	
Eil76	49-24-18-50-25-55-31-10-58-72-39-9-32-44-3-16-63-33-1-73-62-28-74-2-68-6-51-17-40-12-26-67-76-75-4-30-48-29-45-27-52-34-46-8-35-7-53-11-38-65-66-59-14-19-54-13-57-15-5-37-20-70-60-71-69-36-47-21-49-61-22- 64- 42- 43- 41-56-23- 49	
Eil101	20-66-65-71-35-34-78-81-9-51-33-79-3-77-76-50-1-69-27-101-53-28-26-12-80-68-29-24-54-4-55-25-39-67-35-23-56-75-41-22-74-72-73-21-40-58-13-94-95-97-87-2-57-15-43-42-14-44-38-86-16-61-17-84-5-93-85-91-100-98-37-92-59-99-96-6-89-52-18-60-83-45-8-46-47-36-49-64-63-90-32-10-62-11-19-48-82-7-88-31-70-30-20	

To examine the effectiveness of the proposed GA, a test of hypotheses was done to test whether the average best distance found by the proposed GA is significantly lower than a target value (the target value is 1.05 times the best solution ever found for each problem) at significant level  $\alpha$  of 0.05. Table 4 shows the results of this test. The results showed that, in all of the three problems, the proposed GA is significantly lower than the target value.

Table 4. Test of hypotheses on the proposed GA

TS Problem	Hypotheses	Standard deviation	z-Value
Eil51	$\mu_{prop.} = 447.3$ $\mu_{prop.} < 447.3$	0.8815	-91.37
Eil76	$\mu_{prop.} = 564.9$ $\mu_{prop.} < 564.9$	1.2768	-66.13
Eil101	$\mu_{prop.} = 660.5$ $\mu_{prop.} < 660.5$	1.4532	-37.79

To assess the effectiveness of the different mutation types, Table 5 shows the number of times that each mutation type was able to enhance the existing fitness value from one generation to the next one. Also it shows how many times the FC was able to enhance the existing fitness function from one generation to the next one.

Table 5. Number of enhancements for each mutation type

Recombination Strategy	Number of Enhancement
Frequency Crossover	329
EsEm	664
GIm	653
GIm2	2943
REsm	1661
TGsEm	727
REsEm	4609
REm	1412
OPSm	540
MRm	3462

The results showed that REsEm, MRm, GIm2, and REm strategies constitute about 75% of the whole enhancement instances between the generations.

Because these four types constitute 75% of the whole enhancement instances, we would like to examine the effect of removing the other five types from the algorithm. Table 6 shows the results of using just these four types at same average time found in Table 2 (the termination criterion here is to reach same average time found in Table 2).

Table 6. Results for the proposed GA using only four mutation types

TSP	Average best distant for the proposed GA	Best found in all replications	Frequency of best found	Average running time
Eil51	435.35	428.87	12	352 S
Eil76	555.1	545.37	7	451 S
Eil101	659.52	646.64	3	590 S

The results showed that by using the best four mutation strategies the quality of solution was deteriorated in terms of the average distance found and in terms of the frequency of the best solution. This result can be an indication that the exploration and the exploitation of the solution space using these four types mutation only were not as efficient as using all of the nine mutation strategies together.

Table 7 shows the results for using the FC strategy alone. The aim of this test is to see whether the different mutation strategies help in preventing the premature convergence problem or not. The test used the same average time found in Table 2 (the termination criterion here is to reach same average time found in Table 2).

Table 7. Results for using the frequency crossover strategy alone

TSP	Average best distant for the proposed GA	Best found in all replications	Frequency of best found	Average running time
Eil51	1002.36	980.45	1	352 S
Eil76	1486.65	1459.76	1	451 S
Eil101	1956.24	1893.90	1	590 S

In all of the instances the final population was almost identical, which indicates that the population reached a stage of premature convergence and thus trapped in a local optima. It is clear that the average best distance was very bad in all of the three cases. This indicates that the FC stabilizes the population and as a consequence creates a state of premature convergence in the population which halts the evolution process and consequently, deteriorates the fitness value. This results showed clearly that the different mutation strategies help in preventing the premature convergence problem.

Table 8 shows the results for using the mutation strategy only without FC. The aim of this test is to see whether the FC used can help in stabilizing the population and thus can help the algorithm in reaching better solutions. The test used the same average time found in Table 2 (the termination criterion here is to reach same average time found in Table 2).

Table 8. Results for using mutation strategies only

TSP	Average best distant for the proposed GA	Best found in all replications	Frequency of best found	Average running time
Eil51	446.26	442.97	9	352 S
Eil76	580.61	576.95	7	451 S
Eil101	683.12	675.19	5	590 S

In all of the instances the final population was totally diverse and no two chromosomes were identical. The results indicated that the quality of solution found was worse than what was found in Table 2 yet better than Table 7 therefore, we can conclude that the FC used can help in stabilizing the population and thus can help the algorithm in reaching better solutions. This result along with the previous result of Table 7 emphasize that both parts of the proposed algorithm, FC and the nine mutation types, are important and affect the quality of results reached by the proposed algorithm .

## 5. Conclusions

The results showed that both diversity and stability inducers, the nine mutation types and the FC respectively, were essential for the proposed GA to perform well. The diversity inducer reduced the chance of premature convergence and therefore reduced the chance that GA will be trapped in a local optima while the stability inducer stabilized the population so the proposed GA can benefit from the information accumulated over generations in the chromosomes about the optimal solution.

While this GA was applied on TSP, it is believed that it is applicable on any problem that has an Order-Based chromosome representation.

## Acknowledgements

The author is grateful to the Applied Science Private University, Amman, Jordan, for the financial support granted to this research (Grant No. DRGS-2012-2011-57.)

## References

- Abu Bakar, S., Ramlan, M., & Muhammad, S. (2007). Reducing Premature Convergence Problem through Numbers Structuring in Genetic Algorithm. *IJCSNS International Journal of Computer Science and Network Security*, 7(4), 215-217.
- Brady, R. M. (1985). Optimization Strategies Gleaned from Biological Evolution. *Nature*, 317, 804-806. <http://dx.doi.org/10.1038/317804a0>
- Goldberg, D. E., & Lingle, Jr., R. (1985). Alleles, Loci and the TSP. In J. J. Grefenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms and Their Applications* (pp. 154-159). Hillsdale, New Jersey: Lawrence Erlbaum.
- Kureichick, V., Melikhov, N., Miaghick, V., Savelev, A. N. V., & Topchy, P. (1996). Some New Features in the Genetic Solution of the Traveling Salesman Problem. In I. Parmee & M. J. Denham (Eds.), *Adaptive Computing in Engineering Design and Control 96(ACEDC'96)*. 2nd International Conference of the Integration of Genetic Algorithms and Neural Network Computing and Related Adaptive Computing with Current Engineering Practice, Plymouth, UK, March 1996.

- Larra, P., Kuijpers, C., Murga, H., Inza, I., & Dizdarevic, S. (1999). Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. *Artificial Intelligence Review*, 13, 129-170.
- Oliver, I. M., Smith, D. J., & Holland, J. R. C. (1987). A Study of Permutation Crossover Operators on the TSP. In J. J. Grefenstette (Ed.), *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference* (pp. 224-230). Hillsdale, New Jersey: Lawrence Erlbaum.
- Popovic, D. (1997). Retaining Diversity of Search Point Distribution Through a Breeder Genetic Algorithm For Neural Network Learning. *IEEE International Conference on Neural Network*, 1, 495-498.
- Qu, R. (2002). *Case-based Reasoning for Course Timetabling Problem*. Ph.D Thesis. Department of Computer Science, University of Nottingham UK.
- Reinelt, G. (1995). *TSPLIB95*. Universitat Heidelberg.
- Ursem, U. K. (2002). Diversity-guided Evolutionary Algorithms. *Lecture Notes in Computer Science*, 2439, 462-471. [http://dx.doi.org/10.1007/3-540-45712-7\\_45](http://dx.doi.org/10.1007/3-540-45712-7_45)