**IJCTA**, 9(3), 2016, pp. 1553-1562


# Development of a Novel Gene Repair Operator for Genetic Algorithms for Permutation Problems

**R. Lakshmi* and G. Kumaravelan****

***Abstract:*** In this paper, it is presented the outcome of experiment to evaluate the effectiveness of a novel adjunct genetic operator, a Fast Order Mapped Crossover (FOMX). This operator is developed to correct invalid sequences which may be generated following recombinant operators in the Genetic Algorithms (GAs). This paper analyses the consequences of multi point crossover operator on permutation encoded GA, various parameter settings to improve the performance of simple GA, the consequences of replication of genes in a chromosome and the impact of FOMX on permutation based encoding. Using a new gene repair operator FOMX along sides traditional crossover and mutation operators a significant positive side has been able to explore the search space of a problem and generate very good results in an extremely efficient manner, in both time and number of evaluations required. No loss in diversity, convergence speed and in the quality of solution is obtained.

***Keywords:*** Genetic Algorithm, Permutation, Fast Order Mapped Crossover, TSP, Gene Repair, Optimization, Mutation Operators

## 1. INTRODUCTION

Genetic Algorithm is an approximate and optimizing algorithm which is based on the biological evolution process to find the optimal solution in a short span of time. The performance measurements of standard Genetic Algorithm are diversity in the population, convergence speed and the quality of optimal solutions. These three performance characteristics collectively depend on suitable Genetic Algorithm, application of appropriate genetic operators and their settings while solving any optimization problems. However the contribution of crossover and mutation also called as recombinant operators to improve the performance of simple GA is considerable. The crossover operators are classified into three categories, namely binary crossover [1], crossover for permutation encoded chromosomes and gene repair crossover operators.

It is realized from the literature study [2][3] that the impact of any genetic operators and their settings on binary chromosome is not creating any problems. Moreover, it is easy to implement. When the recombinant operators are applied to a permutation coded chromosomes, many issues are emerged. The first issue of applying crossover operators to the permutation coded chromosomes is causing bad disruptions [4] in a chromosome. Hence it destroys the building blocks that are existed in the chromosomes which reduces the fitness values of those chromosomes. The performance of the simple GA depends mainly on the fitness of the individuals and to increase the fitness of all individuals, a penalty function [5] has been chosen. The drawback of the penalty function is designing the penalty function with respect to the problem is a hectic process. It leads to additional computation overheads.

In many problems, Genetic Algorithms may have a tendency to converge towards local optima. This means the large number of individuals sharing the same genetic materials without exploring all search spaces results premature convergence problem. Premature convergence [6] is a loss of genetic diversity of

* Assistant Professor, Department of Computer Science, Pondicherry University, *Email: Karaikalrlakshmiselva@yahoo.co.in*

** Assistant Professor, Department of Computer SciencePondicerry University, Karaikal, *Email: gkumaratcsbdu@gmail.com*

the whole population. This problem may be alleviated by using an appropriate crossover technique that maintains a diverse population of solutions.

This paper redresses the most important issues in performing the crossover operation on permutation encoded chromosomes which leads to duplicate genes existence in their children. In general, multi point crossover operation on permutation coded chromosomes causes more duplicate genes which lead to bad disruptions. An additional gene repairing mechanism [7] is required to correct these invalid chromosomes to valid chromosomes. It works at the genetic level to remove duplicate genes in a chromosome, which considerably slows down the genetic process. Moreover, these kinds of approaches are an additional burden to the evolution of Genetic Algorithms.

As reported in the literature review, most of the existing crossover techniques are compromised with the measurement factors in one way or the other. Some operators increase diversity in the population, but are very slow to converge to the optimal solution and vice versa. Some of the gene repair crossover operators are shown in figure 1. In particular, the experimental results of existing crossover techniques [8][9][10] have been uniformly discouraging. This is because the crossover operators often destroy building blocks of the parent chromosomes. However, it still has quite a low performance, because it does not take into account any information available about the chromosomes to the subsequent generations. The performance of the existing gene repair operators is not remarkable when duplicate genes occur in the new solutions and also there is a lack of gene repair operators available in the field of genetic algorithms.

By taking all these issues into the account and to overcome these drawbacks and limitations, it is necessary to design and develop simple and efficient gene repair operators which have to work quickly and

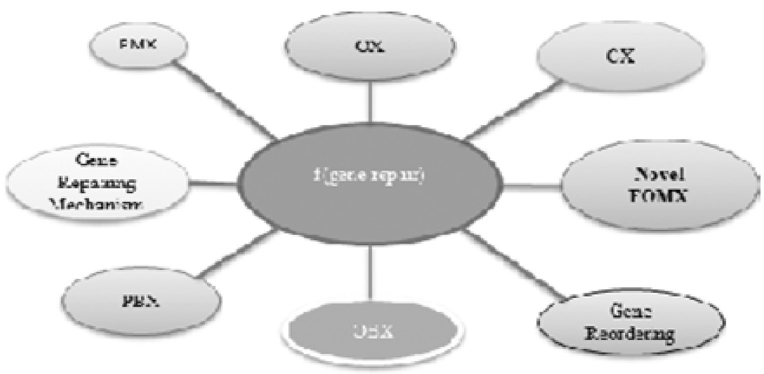


**Figure 1: Types of Adjunct Genetic Operators**

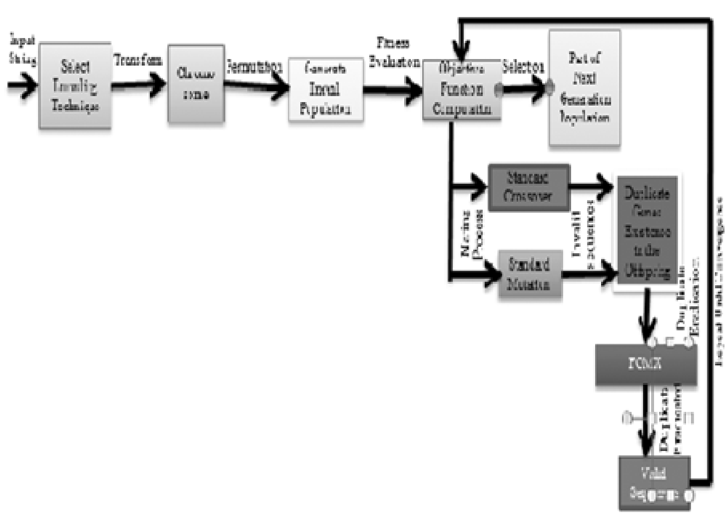


**Figure 2: SGA with Gene Repair Mechanism**

gently. The framework of simple GA with gene repair mechanism is illustrated in figure 2. This approach is applicable to any problem domain where the solution constraints can be identified in the gene string and where the input variables are represented as permutation encoding. This paper explores the very popular types of permutation problems called Travelling salesman Problem (TSP), which is a well-known NP-Complete problem.

## 2. GENETIC ALGORITHM FOR TRAVELLING SALESMAN PROBLEM

TSP [11] is used as a benchmark for many optimization methods. Even though the problem is computationally difficult, a large number of heuristics and exact methods are known to solve the problems. Travelling Salesman Problem is a combinatorial optimization problem where a given set of cities '*C*' and distances '*D*' between any two of them, return the shorted possible valid tour '*T*'. The size of TSP solution space grows very quickly with the problem size. If the number of cities is '*n*' cities, then the number of possible valid tours will be n!. If one uses a technique like brute force to solve the TSP, the computational time is high. When the evolutionary approaches like Genetic Algorithm is used to solve the TSP, the computational time is reduced to some extent. The genetic approach to the TSP [12] relies on designing the chromosome and the fitness function. The fitness value of a chromosome is defined as the cost of each tour or distance of each tour. The objective of this problem is to get the minimum cost of a tour within less time.

TSP involves finding the shortest total distance traveled by salesman. The traditional TSP is represented as follows: Let $G = (V, E)$ be a directed graph where $V = \{0\ldots, n\}$ is the vertex set and $E = \{(i, j): i, j$ € $V$. The objective is to design a network that satisfies all constraints, at the same time minimizing the total travel cost. This model is mathematically formulated and given below:

$x_{i,j}$ € {0, 1}, 0 if there is no path from node *i* to node *j*, and 1 otherwise,

$$i \neq j;\ i, j \text{ € } \{0, 1, 2, \ldots, n\}$$

'*n*' total number of cities

'$c_{i,j}$' cost computed from node *i* to *j*

The objective function for the TSP is

$$Min \sum_{i=0}^{n} \sum_{i \neq i,\, j=0}^{n} c_{ij} x_{ij}$$

This objective function is defined with subject to various constraints and it minimizes the total cost of a tour.

## 3. FOMX METHODOLOGY

Gene repair operators present a solution for order-based problems that uses only standard crossover and standard mutation.

To counteract the invalid chromosomes that occur as a result, introduce gene repair-a genetic repair operator that has a number of positive effects: Gene repair operators examine each chromosome in turn, and do the following:

1. It corrects the number of genes in a chromosome by checking the uniform length of the chromosomes in the population.
2. Removes duplicate genes (repetition) existed in chromosomes
3. No missing genes in a chromosome

Step 1: Select two parents namely P1 and P2

Step 2: Generate two cut points cp1 and cp2 randomly.

Step 3: Swap the elements between cut points of two Parents

Step 4: Store the elements of offspring1 (O1) present within the cut points into the 2-d mapping vector1

Step 5: Store the elements of offspring2 (O2) present within the cut points into the 2-d mapping vector2

Step 6: If any elements of mapping vector1 exist in mapping vector 2 then remove those values in mapping vector1 and mapping vector2

Step 7: Swap the elements of both the offspring according to the values of mapping vector1 and mapping vector2

Step 8: Repeat from step 1 to step 7 for all chromosomes in the population

Step 9: Give birth to the offspring for the next generation

**Figure 3: Pseudocode of FOMX**

Figure 3 gives the pseudocode of the new novel gene repair operator FOMX. It is argued in the literature that the existing crossover operators and gene repair operators in Genetic Algorithms do not alter the genes within the crossover sites after the recombination operation. Instead, it will find duplicate genes outside the crossover region in both left hand side and right hand side of the crossover sites. All genes on both the sides are necessarily compared with the genes existing within the crossover region to check duplication. Since the existing gene repair operators of GAs compare all genes in a chromosome, it takes more, time to converge to get the optimal solution which makes the standard genetic algorithms fail to meet the purpose of it. To effectively increase the convergence speed without compromising the diversity in the population, a FOMX is used.

The working principle of crossover operator on permutation encoded chromosomes is shown in figure 4. The number of chromosomes (tour order) in the population is determined by the population size. Each chromosome is represented as a tour consisting of 'n' number of cities (genes). Here the size of the 'n' is 30 which mean the tour has 30 numbers of cities numbered from 1 to 30.

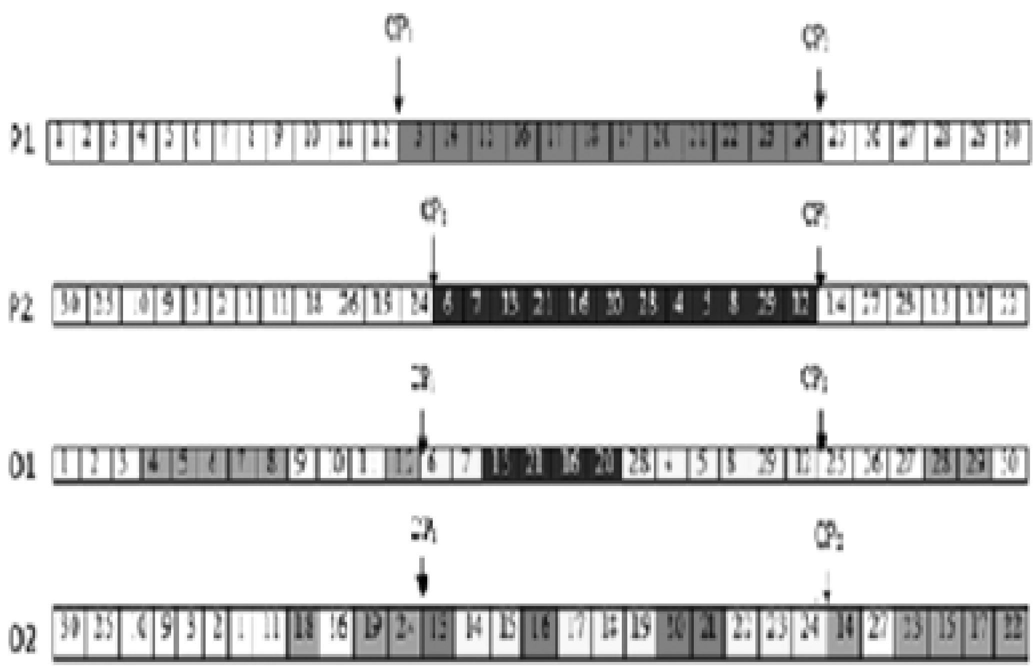


**Figure 4: Two Point Crossover Operation**

After doing the crossover process duplicate genes are existed in the offspring and the same can be removed and repaired by a new novel FOMX operator which is entirely different from the existing crossover and gene repair operators.

The working principle of FOMX operator is shown in figure 5 and figure 6. This operator is applied after the standard crossover operation. FOMX operator does not worry about what elements apart from cut points are. It will not perform any comparison before the cut point cp1 and after the cut point cp2 rather does a comparison within the crossover sites. After swapping substrings (genes in crossover section) of an individual, duplicate genes presented within the cut points are checked and detected. If duplication has occurred, the position of the duplicate genes is stored in the mapping function. According to the mapping function values, elements of offspring are corrected and produced as a perfect individual. This process is continued for all individuals in the population and builds the next generation population.

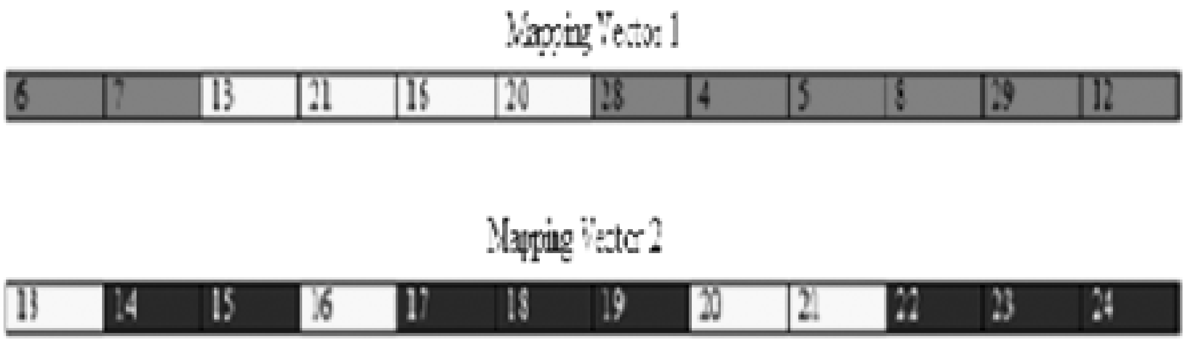


**Figure 5: Values of Mapping Vector**

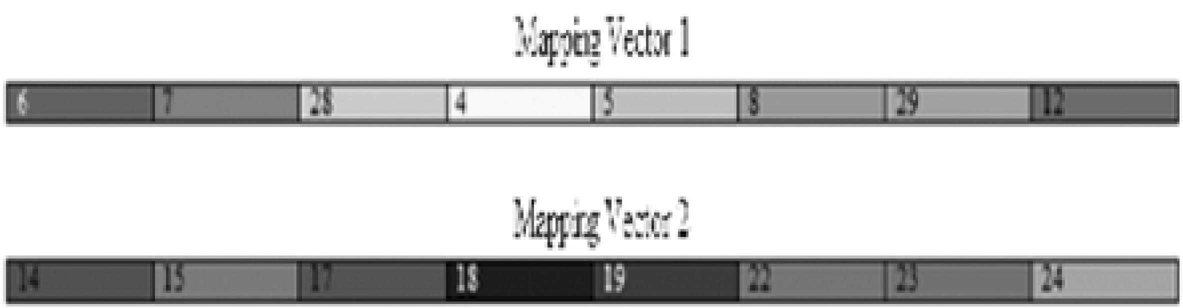


**Figure 6: Mapping Vector after Removing Duplicacy**

## 4. EXPERIMENT–TSP

Some types of encoded chromosomes require specially designed crossover and mutation operators... like the Traveling Salesman Problem in which the task is to find a correct ordering for a collection of individuals. The natural choice of representation for the TSP is an Order-based representation [13]. Additionally, the genetic operators employed must also be order-based. If either the representation or the operators do not respect the order based nature of the problem, then invalid solutions will be generated. Although the TSP is NP-Hard, it may be classified by two separate facets: Optimization and Permutation. Responsibility for optimization lies with the standard genetic algorithm, which effectively remains unchanged from Holland [14]. Responsibility for only allowing valid permutation in the population lies solely with the gene repair operator. The majority of gene repair replacements were performed in a left-to-right manner-replacing the left-most duplicate node first. Additionally, the replacement node was retrieved from the template also in a left-to-right manner. But the FOMX gene repair technique processes only the crossover region which is already explained in section 3.

In our earlier work we looked at the crossover operators that respect the order-based nature of permutation problems, and prevent the introduction of errors such as invalid tours. The order preserving crossover operators that have been tested and analyzed for various TSP instances taken from TSPLIB [15] include: Order Crossover, Modified Crossover, Partially Mapped Crossover, Cycle Crossover, Position based crossover, Greedy crossover, Sorted order crossover plus a number of less frequently used crossover operators.

In this paper again the TSP problem has been tested on three another gene repair techniques namely static template [16], parent based template and random template. The results of these techniques are compared with the result of the novel FOMX operator and thus the performance is analyzed.

## 5. RESULTS AND DISCUSSION

This research analyzes the performance of FOMX and compares it with three gene repair techniques. The various methods, including FOMX mechanism are implemented in MATLAB tool [17][18] on an Intel i5 system. To prove the efficiency of the FOMX operator, few TSP benchmark instances, namely d1291, nu3496, eg7146, gr9882, d15112 and d18512 have been taken up for performance evaluation. The co-ordinate values matrix of all TSP instances are converted into the distance matrix by using the equation which is given below

$$\text{Dist. between two points} = \sqrt{\left\{\left(x2-x1\right)^2+\left(y2-y1\right)^2\right\}}$$

The results have been taken for ten runs. In each run, the selection rate and mutation probability were taken dynamically. This implementation uses different selection rate, crossover rate and mutation rate. The selection rate has been varied from 10 to 15%. The mutation rate has been varied from 5 to 10%. From the experimental results it is clearly understood that the results are better when the selection rate is 15% and the mutation rate is 10%.

The optimal distance of TSP instances eg7146 and gr9882 are also computed.

The optimal distances of various TSP benchmark instances are shown in table 1 to 3. From the results it is clearly known that the FOMX gene repair operator produces very near optimal distances of all TSP benchmark instances.

From the results of the tables 1, 2 and 3, the average error rate [19][20] of the existing gene repair techniques and for the FOMX are calculated and it is shown in table 4. The table 4 also depicts the average computational time of all gene repair techniques. The time taken by the FOMX gene repair operator for the six TSP instances is very minimal when compared to the existing gene repairing mechanism which is also

**Table 1**
**Optimal Distance of TSP Instance D1291**

| *No. of Runs* | *Static Template* | *Random Template* | *Parent Based Template* | *FOMX* |
|---|---|---|---|---|
| 1 | 66601 | 69301 | 63551 | 64301 |
| 2 | 68651 | 63451 | 62301 | 66776 |
| 3 | 69086 | 63251 | 64801 | 67621 |
| 4 | 70811 | 62186 | 72051 | 63796 |
| 5 | 61821 | 69801 | 69551 | 64911 |
| 6 | 64101 | 63401 | 69551 | 72121 |
| 7 | 65996 | 60301 | 69551 | 66761 |
| 8 | 67441 | 70051 | 65951 | 65186 |
| 9 | 70096 | 71801 | 71801 | 62231 |
| 10 | 65996 | 67801 | 65951 | 60601 |

**Table 2**
**Optimal Distance of TSP Instance D15112**

| *No. of Runs* | *Static Template* | *Parent Based Template* | *Random Template* | *FOMX* |
|---|---|---|---|---|
| 1 | 2044834 | 2044834 | 1921684 | 2050309 |
| 2 | 2211334 | 2355667 | 2164584 | 1901074 |
| 3 | 2039284 | 2567842 | 1953884 | 1950084 |
| 4 | 2044834 | 2056782 | 2076784 | 1962934 |
| 5 | 1989334 | 2314586 | 1963684 | 1963184 |
| 6 | 1989334 | 2266879 | 2004884 | 2028184 |
| 7 | 1989334 | 1985252 | 2041734 | 2060284 |
| 8 | 2133634 | 2234211 | 2114884 | 1886734 |
| 9 | 2039284 | 2133421 | 2160484 | 1617483 |
| 10 | 2133634 | 2123125 | 2056684 | 1604918 |

**Table 3**
**Optimal Distance of TSP Instance D18512**

| *No. of Runs* | *Static Template* | *Parent Based Template* | *Random Template* | *FOMX* |
|---|---|---|---|---|
| 1 | 785488 | 900656 | 809038 | 782588 |
| 2 | 834988 | 917488 | 833738 | 793308 |
| 3 | 800638 | 914188 | 846838 | 837798 |
| 4 | 785488 | 818828 | 893638 | 777787 |
| 5 | 783988 | 800638 | 859438 | 791318 |
| 6 | 783988 | 798688 | 833938 | 806368 |
| 7 | 783988 | 894567 | 822688 | 807758 |
| 8 | 756338 | 876354 | 838438 | 695688 |
| 9 | 800638 | 824563 | 917488 | 669338 |
| 10 | 811888 | 808761 | 942838 | 686738 |

**Table 4**
**Analysis W. R. to Error Rate and Computational Time**

| *Crossover Method* | *Parameters* | *ISP Benchmark Instances* | | | | | |
|---|---|---|---|---|---|---|---|
| | | dl291 | nu3496 | eg7146 | gr9882 | dl6112 | dl8512 |
| | Re Err (%) | 25629 | 25839 | 24133 | 26.52 | 22.16 | 25.85 |
| Random Template | Ac. Err (%) | 32.987 | 29.217 | 31.289 | 33.939 | 30.058 | 33.254 |
| | Ave.time | 4 | 24 | 72 | 96.3 | 168 | 269 |
| | BeErr (%) | 24.380 | 4.566 | 35.44 | 24.45 | 26.201 | 23.181 |
| Parent BasedTemplate | Ac. Err (%) | 29.490 | 14.735 | 50.771 | 39.331 | 40.377 | 32.582 |
| | Ave.time | 3.53 | 25 | 50 | 98 | 169 | 260 |
| | BeErr (%) | 20.078 | 22313 | 21.607 | 21.106 | 22.684 | 18.666 |
| Static Template | Ac. Err (%) | 29.6651 | 32.451 | 32.02 | 31.918 | 33.840 | 33.628 |
| | Ave.Time | 3.45 | 22 | 48 | 95.3 | 166.5 | 255 |
| | BeErr (%) | 19.29 | 1.841 | 4.816 | 8.275 | 2.023 | 3.735 |
| FOMX | Ac. Err (%) | 28.797 | 13.718 | 23.116 | 24.883 | 20.941 | 18.140 |
| | Ave. time | 3.6 | 12 | 43 | 51 | 83 | 103 |

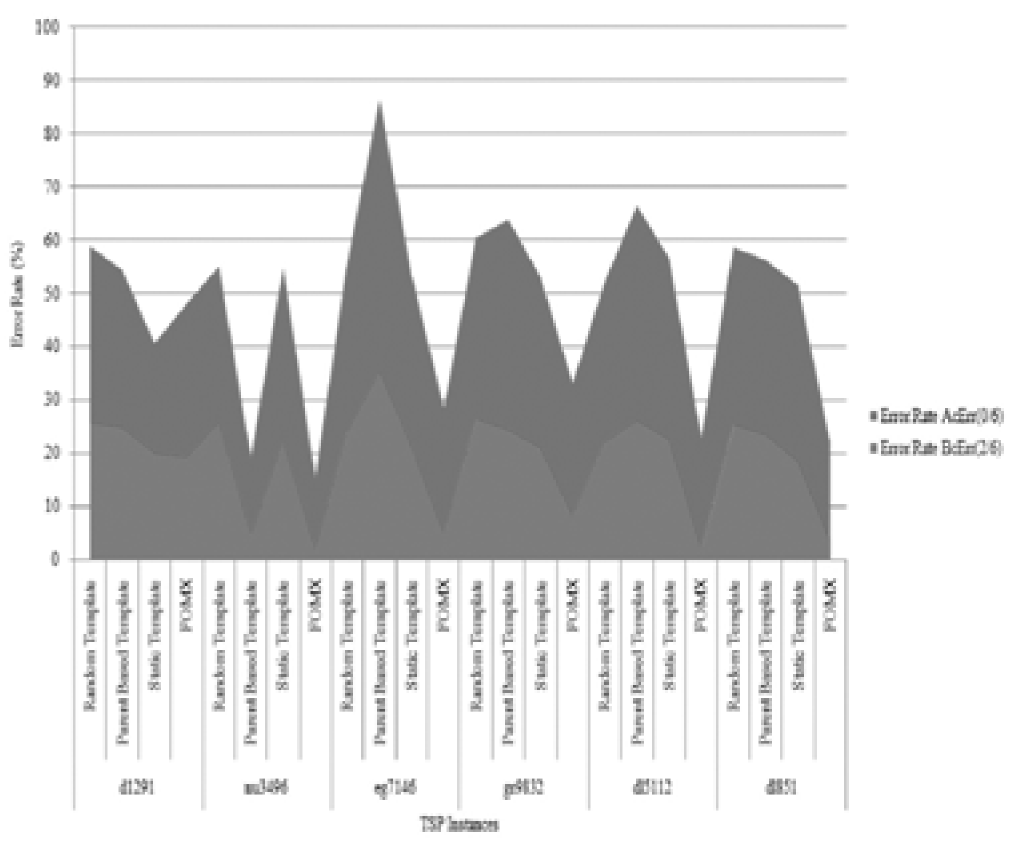


**Figure 7: Error Rate Range of Existing Gene Repair Techniques and FOMX**

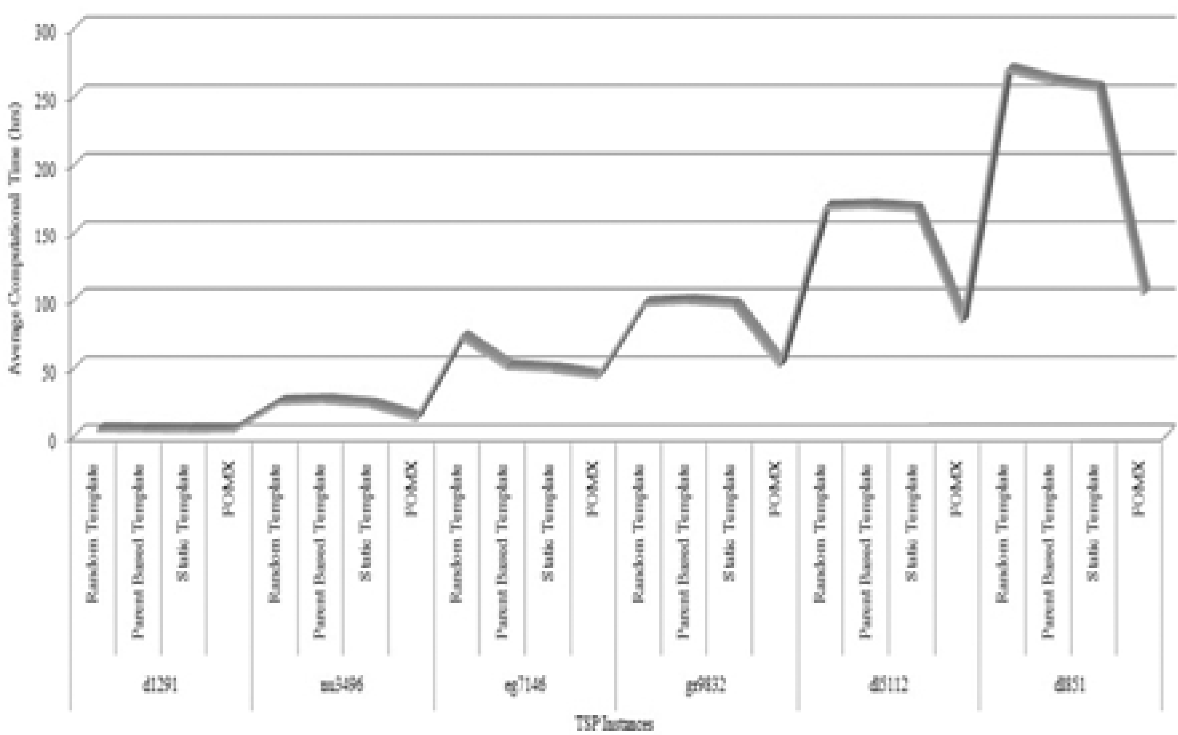


**Figure 8: Average Computational Time of Existing Gene Repair Techniques and FOMX**

shown in the performance graph figure 8. From the figure 8, it is evidently proved that the novel FOMX operator outperforms the existing crossover operators in terms of computational speed.

## 6. CONCLUSION AND FUTURE WORK

The performance of the standard GA is based on the fitness of the chromosomes. The fitness of the chromosomes can be improved by the genetic recombinant operators and the selection of the relevant Genetic Algorithms. Though there are several methods existing, there is a quest for improvisation of these recombinant operators. One such attempt has been made and devised a gene repair operator FOMX. The benefits of FOMX were evaluated in depth on different sizes of TSP problem. A detailed analysis of FOMX and existing gene repair techniques have been made experimentally also. It explored the use of gene repair on the TSP using the fitness function to optimize the solution while gene repair ensures the validity of solutions. Results produced by FOMX operator have either reached global optimal solutions, or have been close to optimal solutions. Furthermore, solutions appear in a relatively small number of generations which in turn increases the convergence speed of a simple GA. Future work is necessary to compare the effectiveness of novel FOMX gene repair operator against the other permutation problems like Vehicle Routing Problem and also against the order-preserving mutation operators.

### *Acknowledgment*

The suggestions and motivation of Dr. K. Vivekanandan, Professor of computer science, Pondicherrry Engineering College, Puducherry, India is gratefully acknowledged

### *References*

[1] R. Lakshmi, Dr. K. Vivekanandan, “An Analysis of Recombination Operator in Genetic Algorithms”, IEEE International Conference on Advance Computing (ICoAC 2013, MIT, Chennai, India.

[2] Davis L. “Handbook of Genetic Algorithms” Van Nostrand Reinhold, 1991.

[3] Goldberg, D. E., “An analysis of a reordering operator with tournament selection on a GA-hard problem”, Lecture Notes in Computer Science (LNCS), 2723, pp. 825–836, 2009.

[4] Arthur L. Corcoran, Roger L. Wainwright, “Reducing disruption of superior building blocks in genetic algorithms”, Proceedings of the, ACM SIGAPP Symposium on Applied Computing February,

[5] Mustafa Kaya, “The effects of two new crossover operators on genetic algorithm performance. Applied Soft Computing”, 2011 Elsevier, Vol. 11 pp. 881–890.

[6] Alfonsas Misevičius, Bronislovas Kilda, “Comparison of Crossover Operators for the Quadratic assignment problem”. Journal of Information Technology and Control, 2005, Vol. 34, ISSN: 1392–124x

[7] Dunwei GongRcga, and et al, “Adaptive interactive genetic algorithms with individual interval fitness”, 2007 Elsevier, pp. 359-365.

[8] Al jaddan, O., Rajamani, L., and Rao, C.R., “Improved Selection Operator for GA”,. Journal of Theoretical and Applied Information Technology, 2005, pp. 269–277.

[9] Lakshmi and Dr. K. Vivekanandan, “Heuristics Based Learning on Human Psychology”, IJCA, ISSN 0975–8887, Feb 2013.

[10] Qingfu Zhang, and et al, “An Evolutionary Algorithm with Guided Mutation for the Maximum Clique Problem”, IEEE Transactions on Evolutionary Computation, 2005, Vol. 9, Issue 2.

[11] P. Larra˜ naga, et. Al, “Genetic algorithms for the Travelling Salesman Problem: A Review of Representations and Operators”, Artificial Intelligence Review, 1999.

[12] R. Laksmi and K. Vivekanandan, “A Novel Methodology for Genetic Algorithms in Crossover Operation: Segment Replacement Operator”, Alan Piszcz, Terence Soule, “A Survey of Mutation Techniques in Genetic Programming”, ACM, 2006, 59593-186.

[13] Naveen kumar, J.U., and et al, “A Comparative Analysis of PMX, CX and OX Crossover operators for solving Travelling Salesman Problem”, International Journal of Latest Research In Science and Technology (IJRSAT), 2012, Vol. 1, issue 2, pp. 98-101, ISSN: 2278-5299.

[14] Oliver, I.M., Smith, D.J. and Holland, J.H, "A study of permutation crossover operators on the travelling salesman problem", 1995, Proceedings of the Third International Conference on Genetic. Roy. Soc. London, vol. A247, pp. 529-551.

[15] www. Tsplib//

[16] George G. Mitchell et. Al, "GeneRepair – A Repair Operator for Genetic Algorithms.

[17] MathWorks, Genetic Algorithm Toolbox User's Guide for MATLAB.

[18] Ray, SS., Bandyopadhyay, S., Pal, SK, "New Operators of Genetic Algorithms for Traveling Salesman Problem", Evolutionary Computation 2004, pp. 1457-1464.

[19] Andrej Kazakov, "Travelling Salesman Problem: Local Search and Divide and Conquer working together", Independent Research Review, University of Southmapton, 2011.

[20] Victor Paul, P., and et al., "A New Population Seeding Technique for Permutation Coded Genetic Algorithm: Service Transfer Approach", 2013, Journal of Computational Science, Elsevier.