Advances and Applications in Mathematical Sciences
Volume 17, Issue 1, November 2017, Pages 1-29


# A NOVEL SELF-ORGANIZATION MODEL FOR IMPROVING THE PERFORMANCE OF PERMUTATION CODED GENETIC ALGORITHM

**K. DINESH[1], J. AMUDHAVEL[2] and R. SUBRAMANIAN[3]**

[1,3]Department of Computer Science and Engineering
Pondicherry University, Puducherry, India
E-mail: dineshhumar@gmail.com
rsmanian.csc@pondiuni.edu.in

[2]Department of Computer Science and Engineering
K.L. University, Andhra Pradesh, India
E-mail: info.amudhavel@gmail.com

## Abstract

Genetic Algorithms (GA) are extremely dominant problem solving strategy, using the power of evolutionary principles and being used in variety of fields for solving more complex problems. Varieties of assistive techniques have been proposed to improve the performance of Genetic Algorithms w.r.t. the nature of the application and Self organization is one such model, which is aimed at improving the performance of the GAs by all means. The Self-organization models enable the systems to acquire and maintain the structure by themselves, without any external control. It is highly evidenced that it gives greater benefits to solve the complex problems with competent efficiency levels in conjunction with the classical GAs. The combined version of SOM and GA has the power of better exploration and so possess high probability of finding many local optima in addition to the global optimum. In this way, the work reported in this paper proposes an efficient self-organization model for improving the performance of the GA in a different perspective. Here, the Travelling Salesman Problem (TSP), a popular NP-hard problem is being chosen as the test bed and the standard benchmark TSP instances, which are obtained from TSPLIB [33] are considered as the test data. The competency of the proposed model is demonstrated by means of a set of well-defined experiments over the selected benchmark TSP instances. The assessments proved the efficiency of the technique in terms of a set of generic performance criteria like convergence rate, convergence time, error rate, nearest neighbor ratio and distinct individuals.

2010 Mathematics Subject Classification: 68N13, 97N50.

Keywords: Self-organization technique, Genetic algorithm, population seeding technique, traveling salesman problem, pattern replacement, combinatorial problem.

Received March 10, 2017; Accepted July 20, 2017

## 1. Introduction

It is widely demonstrated that the Genetic Algorithms (GA) are extremely dominant problem solving strategy, using the power of evolutionary principles and have been used in variety of fields to solve more complex problems. The objective of the GA is to find the optimal solution which is near to the objective function of the optimization problems with very large search space [21]. The classical GA consists of several phases and among those, initialization phase is to be considered as very important, where the quality of the initial solutions at the starting point is to be decided. Traditionally, the population initialization is performed by random initialization methods, which may also produce larger quantity of worst individuals [29, 45, 54]. On the other hand, induced models of population seeding techniques lead the belated achievements of the fitness functions with poor convergence values [26, 30, 50]. This tradeoff motivated the researchers to find new sets of population initialization techniques for improving the phase specific as well as overall performance of the family of the GAs. In this perception, new mixture of initialization methods that combine the benefits of both random and induced initialization has been proposed by many of the researchers.

Variety of proposals regarding population seeding techniques for GA encouraged the users to select the appropriate model w.r.t. the nature of the application. At the same time, it also increases the volume of the repository of such category, which resulted in ambiguity in decision making. This vagueness motivated the researchers in a different direction in order to find appropriate techniques to improve the efficiency of the existing set of initialization techniques rather than proposing new techniques. Self-organization is one such model, which is aimed at improving the performance of the GAs. Self-organization often defined as global order emerging from local interactions appropriately [15, 28]. Self-organization is a dynamic process and adaptive process where the systems acquire and maintain the structure by themselves, without any external control. It is highly evidenced that it gives greater benefits to solve the complex problems with competent efficiency levels. In self-organizing, individuals are interacting directly with the candidate solutions to produce the common pattern which influences the behavior of low level individual in the population [15, 17, 40]. These systems can scale upto the enormous size since the interaction is subsequent local individuals rather than the global medium. The multi-entity self-organization system proposed by Bak [11] selects the worst fitness individuals in each iteration and updates its position according to the local interaction. This method also affects their neighborhood individuals. Consequently, the local fitness of the individuals will be increased due to the update method influenced by the local interaction among the individuals.

In the view of evaluating self-organizing ability, the argument of assessing autonomous agents w.r.t. local fitness function was introduced by Han (2003). The framework for Autonomy Oriented Computing was proposed by Liu et al. (2005; 2006). It uses self-organized distributed autonomous agents for characterizing complex systems behavior of the optimization problem in the intended domain. An unique model of self-organization for GA called GASOM, which stands for "Genetic Algorithm using Self-Organizing Map" was proposed in [12, 27].

Self-Organizing Maps (SOM) is mainly an unsupervised learning mechanism which is used to avoid the local optima and premature convergence of GA. The combined version of SOM and GA has the possibilities to find the many local optima instead of one global optimum. SOM holds well in non-linear mapping for large search space. The Hopfield-type network and Kohonen types of SOM are used to solve NP-hard TSP problem [1, 3, 6, 14]. Hopfield-type network has its major focus to solve small and medium scale problem, whereas the Kohonen type of SOM focused on large size problem space. Many types of SOM algorithms [7, 18, 19, 22, 39 41, ] have been proposed to solve the Travelling Salesman Problem (TSP) by incorporating the efficient initialization method and parameter adoption. Hui-Dong and Kwong-Sak Leung [39] presented two other versions of SOM: ISOM-Integrated SOM and ESOM-Expanding SOM.

These techniques are used to solve the TSP problem, which required human intervention resulted in increased complexities in design of solutions. Kwon Jeong and Jang lee [23] proposed the multimodal function optimization, which offered an improved performance by adapting the parameters GA. A new dominant selection technique, which selects the dominant individuals and cyclical mutation operator, has been proposed in [24]. According to dominant individuals and cyclical mutation operator, mutation probability will be tuned periodically. Although it offers better optimization result, it suffered by faster convergence rate, which lead to end up with premature convergence. As another development, self-organizing algorithm (migration Algorithm) and genetic algorithm are combined in order to avoid the premature convergence and get trapped from the local optimum [17, 24, 44]. In random immigrant technique, candidate individuals interact between themselves and find the individual near to optimal solution and placed in subpopulation. The combination of self-organizing (migration Algorithm) and genetic algorithm gives different parameter design and setting such as population size, crossover probability and mutation probability [7]. In its series, a new replacement technique has been proposed in [41] that replaces worst individual of the population by the random immigrants. This random immigrant increases the individual diversity among the population through random replacement of individuals in the population. But the random fashion of selecting the replacement individuals suffers from low fitness values. In general, in most of the cases [17, 20, 38, 46, 25], it is observed that self-organization models for GAs are proposed and proved by applying the hybrid models to solve the Travelling Salesman Problem (TSP), which is usually belongs to the NP-hard problem category. In this way, the work reported in this paper proposes an efficient self-organization model for improving the performance of the GA, which is also demonstrated by applying the proposed model for the standard benchmark TSP instances.

The organization of this paper is as follows: Section 2 offers the background information about various population initialization techniques, which are used in this work and the proposed model. Section 3 describes the algorithm for proposed model and population initialization techniques used in this paper. Section 4 reports about the experiments carried out in this work and the corresponding result analyses in comparison with the existing techniques. At last, Section 5 offers the conclusive remarks over the reported work.

## 2. Background Information

The Initial phase of population initialization plays the important role in providing the good quality individual for the population which improves the optimal solution. The Individuals (Ind) are represented as $(C_1, C_2, C_3, \ldots, C_N)$, which is a permutation of $(1, 2, \ldots, N)$. This section offers the brief introduction about various population initialization techniques that are applied in this paper.

### 2.1. Random Population Initialization

Random population initialization is one of the important techniques in which the individuals are created in random manner without any heuristics [2, 10, 28, 47, 48]. The population generated using random population seeding technique does not suffer by local optima, however it exhibits poor convergence rate when compared with other techniques.

In case of TSP, the random initialization technique normally chooses the first city in a random fashion and repeats the same until the candidate individual is formed. The algorithm for this technique stated in the Figure 2(a). In the random technique, the individual is formed by choosing the first city $C_1$ in the random manner. In order to select the next city $C_j$ where $C_j \notin$ Indi, then this process continues until individual is created.

| Symbols | Explanation |
|---|---|
| N | Number of cities in the Individual. |
| popsize | Maximum number of Individual In eachpopulation. |
| Ind | Individual in the population $(C_1, C_2, C_3 \ldots \ldots, C_N)$. |
| POP | Population in the generation $(C_1, C_2, C_3 \ldots \ldots, C_N)$. |
| Gen | Number of generation $(\mathrm{POP}_1, \mathrm{POP}_2, \ldots, \mathrm{POP}_{\mathrm{Gen}})$. |
| ba | Best adjacent number. |
| bax | Select the value randomly within the range $(1 \leq \mathrm{bax} \leq \mathrm{ba})$. |
| p | Number of Ind as leader Individuals |
| IndL | Leader Individual. |
| IndA | Active Individual. |
| $\mathrm{Ind}_{\mathrm{best}}$ | Individual with best fitness value |
| el_rate | Number of Elitist individual carried to nextGeneration |
| Pc | Crossover probability. |
| Pm | Mutation probability. |
| CityList | List of cities visited by the Ind at current state. |
| $C_{\mathrm{Last}}$ | Recent city visited by Ind. |
| Seq | Sequence discovered from the set IndL |
| r | Size of the sequences |
| s | refers to no. of sequence derived from the set Seq |

**Figure 1.** List of Variables used in the proposed algorithm.

### 2.2. Nearest Neighbour (NN) Population Initialization

The nearest neighbour population initialization technique is one of the efficient techniques, which has the increased convergence rate [31, 38]. In this case, the individuals are created by visiting the nearest city from the last city visited by any individual. Although this technique may be preferred with better convergence rate, it has the drawbacks of poor diversity and trapping with the local optima.

The algorithm of *NN* population technique is formulated as an algorithm and it is clearly illustrated in the Figure 2(b). In the nearest neighbor technique, the initial population for the TSP is created using the by selecting the first city $C_1$ in the random way, then remaining city $C_j$ is selected by choosing the next nearest city from the $C_{Last}$ where $C_j \notin \text{Ind}_i$ then visit $C_j$. The initial population POP is created using the same process.

The algorithm of *NN* population technique is formulated as an algorithm and it is clearly illustrated in the Figure 2(b). In the nearest neighbour technique, the initial population for the TSP is created using the by selecting the first city $C_1$ in the random way, then remaining city $C_j$ is selected by choosing the next nearest city from the $C_{Last}$ where $C_j \notin \text{Ind}_i$ then visit $C_j$. The initial population POP is created using the same process.

### 2.3. ODV Based VV Population Initialization Neighbour

The Ordered Distance Vector based *VV* (Vari-begin with Variable diversity) technique is one of the recently proposed techniques proposed in 2013 [10]. This ODV method has the advantage of simplicity, randomness and individual diversity, which are mandatory for all population seeding techniques.

This algorithm is described in the Figure 2(c). Initial population is created by using the Vari-begin with Variable diversity (*VV*) by Order Distance Vector (ODV). In *VV* (ODV) population initialization method, first city is generated randomly to all the individuals in the population. Generate the bax number randomly generated from the Best Adjacent $(1 \leq \text{bax} \leq ba)$ and select $C_j$ from the ODM by using the bax and recent city visited by the individual. If $C_j$ is not visited by the individual then visit $C_j$. If all the best adjacent cities are visited by the individual, then randomly generate $C_j$ where $C_j \notin Ind_i$ then visit $C_j$. This process continues until all the individuals for the population (POP) are generated.

### 3. Proposed System - Self Organization Algorithm

This work uses the common pattern replacement model to construct a simple self-organization algorithm, which is combined with the classical GA. This combination avoids the premature convergence and improves the diversity rate of the overall GA. In this self-organization algorithm, the candidate individuals of any population are divided into two categories: individuals with high fitness values are called as Leader individuals (*IndL*); remaining individuals are termed as Active individuals (*IndA*). This algorithm doesn't use any more complex heuristics for defining the threshold for high fitness value and reaming. Here, in each population, the top ranked individuals are marked as the set of *IndL* and the remaining are marked as *IndA*. After this, all other individuals will be made to move towards the leader individuals by incorporating the intended sequence (common pattern) found from the leader individuals. With respect to the TSP, the fitness of each individual (Ind) in a population set is function

Fitness: $POP \rightarrow \mathbb{R}^{+}$ given by

$$\text{Cost of the cycle} = F(Ind_j) = \sum_{i=1}^{N-1} C_{i,\, i+1} + C_{N,1} \forall Ind_j \in POP, \tag{1}$$

where,

- $N$ refers to the number of cities in the individual,
- $C_{i,\, i+1}$ refers the distance between two cities $i$ and $i+1$,
- $C_{N,1}$ refers to the distance between last city and first city during return after the tour.

Let POP be the raw set of individuals for a population set and it is defined as,

$$POP = \{IND_1,\ IND_2,\ IND_2,\ \ldots,\ IND_{popsize}\}, \tag{2}$$

where, popsize refers to the number of individuals in the population set.

After sorting the elements in POP based on their fitness function $F$, we get

$$POP_{sorted} = \{Ind_1, Ind_2, \ldots, Ind_{popsize}\} \tag{3}$$

such that $F$ is monotonically increasing function for $i = 1$ to popsize. Then the individuals from the $POP_{sorted}$ is divided into two subset based on the fitness value of the individuals. We drive two subsets $IndL$ and $IndA$ such that

$$IndL \cup IndA = POP_{sorted} \tag{4}$$

$$IndL \cap IndA = \varnothing \tag{5}$$

$$| IndL | \leq | IndA | \tag{6}$$

$$| IndL | = p. \tag{7}$$

Here, $IndL$ is the set of top $p$ ranked individuals with high fitness (minimum distance) values is called as Leader individuals and $IndA$ is the set of remaining individual from the $POP$ with less fitness value (maximum distance) values is called as Active individuals and $p$ is the cardinality of the set $IndL$. Further the set $IndL$ has been utilized for generating the continuous common sequence which is incorporated in the set $IndA$ in order to improve the fitness values of the individuals.

Common sequence $G$ is defined as,

$$G = \xrightarrow[\cap]{\text{Order of Visit}} \text{form } A \times B \to C$$

such that

$$G(x, y) = z, \tag{8}$$

where, $z$ is the largest common continuous subsequence of $x$ and $y$, $\forall x \in A$; $y \in B$.

The common sequence/pattern may be defined as the set of the similar elements with identical order in two individual from the set $IndL$. It is described as follows:

$$Seq_{i,j} = G(Ind_i, Ind_j), \forall Ind_i, Ind_j \in IndL \text{ and } i \neq j. \tag{9}$$

Finally, the set *Seq* would be as $seq = \{C_i, C_{i+1}, C_{i+3}, \dots, C_v\}$, where $v \leq N$ refers to the maximum length of the *Seq* or the number of elements in the set *Seq*. For example, let $IndL_i$ and $IndL_j$ are two distinct individual with the permutation of $\{C_i, C_{i+1}, C_{i+3}, \dots, C_N\}$ from the set of *IndL*. The sequence (*Seq*) discovered from the leader individual is the set of cities visited in same order by $IndL_i$ and $IndL_j$. The $Seq_{i,j}$ obtained from two distinct individuals $Ind\,L_i$ and $IndL_j$ is $\{C_i, C_{i+1}, C_{i+3}, C_{i+4}, C_v\}$. It is also observed that, in the proposed common pattern replacement model, the process of replacement as a whole with higher values of '$v$' lead to struck with local optima. Empirical analyses proved that the replacement process in terms of sub patterns rather than the whole pattern produces good results by minimizing the probability of trapping with local optima. It is observed from the [34] that the best adjacent value for the TSP is 4. Then it has been utilized for the sub-patterns with the length of 4 yields better results. Hence, based on the empirical analyses, for all experiments reported in this paper, the sub-pattern length is maintained as 4 and in this view, the Equation (9) can be defined as,

$$Seq_{i,j}^{s,r} = \left\{\frac{Seq\,(i,\ j)}{|\ Seq\,(i,\ j)\,|} = r = \{(C_i,\ C_{i+1},\ C_{i+3},\ C_r),\ \{C_{i+1},\ C_{i+3},\ C_{i+4},\ C_v\}\}\right\} \quad (10)$$

where,

- $s$ refers to the number of sub patterns derived from the set *Seq*.
- $r$ refers to the size of each sub-pattern, which is fixed as 4.

Consider an individual from the set $IndA$ as $Ind_k$. Here, by considering the subset $Seq_{i,j}^{s,r}$ as in the Equation (10), the identical sub-sequence of $MSeq$ is to be found in $Ind_k$, such that the length of the sub-sequence should be always equal to $r - 1$.

In general, the sub-sequence $\{C_i, \dots, C_{r-1}\}$ of $seq_{1,r}$ is to be found in the individual $Ind_k$ and it is described as follows:

$$MSeq = G(Seq_{i,j}^{s,r-1},\ Ind_k),$$

where

$$Seq_{i,j}^{s,r-1} \in Seq;\ Ind_k \in IndA. \tag{11}$$

Then, the last element of the subset $C_r$ should be identified in $Ind_k$ along with its position. Here a heuristic must be applied for finding the appropriate location for finalizing the common. The position of $C_r$ must be changed such that it should be adjoined with the available sub-sequence, such that the sequence of $\{C_i, C_{i+1}, C_{i+3}, C_r\}$ is present in $Ind_k$ also in the same order.

The general form of heuristics can be stated as follows: The r$^{th}$ element of $Seq_{i,j}^{s,r}$ should be identified in $Ind_k$ and it should be adjoined with sub-sequence such that, the subset *MSeq* is available in $Ind_k$ in the same order of visit. In the above heuristics, both the cases must be executed and based on the fitness value one of the cases may be considered and continued. i.e. in our example, the execution with minimum distance value may be considered and continued with that. Similarly, the process may be repeated for all other subsets of $Seq_{s,r}$ until the completion of all the individuals in the set $IndA$. It is also observed that the sub-pattern replacement policy offers extended possibilities to incorporate the leader sequences into the active sequences and also avoid the problem of premature convergence.

**Assumption**

— Let $p$ be the number of top ranked individuals of the set IndL and it is fixed as as 10.
— Let *el_rate* number of elitist individuals carried to next Generation and it is fixed as 4.
— The crossover probability Pc value is fixed as 0.7.

**Self-Organization Algorithm**

Step 1: Initialization:

1.1. Choose the appropriate TSP instance from TSPLIB.
1.2. Generate the initial population set using anyone of population initialization technique.

Step 2: Evaluate the fitness of each individual using the Eq.(1).

Step 3: Get the sorted population set $POP_{sorted}$ from Eq.(3).

Step 4: Derive the sets $IndL$ and $IndA$ as described in the Eqns. (5) and (6).

Step 5: Find the common sequence in $IndL$ from Eq.(10).

Step 6: Move the $Ind$A towards $IndL$ from Eq.(11).

Step 7: Rebuild the population by combining $IndL \cup IndA$.

Step 8: Evaluate the fitness of each $Ind$ in the $POP$ using Eq.(1) and sort population with respect to fitness value using Eq.(3).

Step 9: Select the elitist Individuals according to the *el_rate* to the next generation.

Step 10: Apply greedy crossover for $POP$ with crossover probability *Pc.*

**Figure 2.** Proposed Self-Organization Algorithm for GA.

## 4. Algorithm Explanation

The population initialization technique plays an important role in providing the good quality individual for the population which improves the optimal solution. The Individuals $(Ind)$ are represented as $(C_1, C_2, C_3, \ldots, C_N)$ cities, which is a permutation of $(1, 2, \ldots, N)$.

In $VV\,(ODV)$ population initialization method, first city is generated randomly to all the individuals in the population. Generate the $bax$ number randomly generated from the Best Adjacent $(1 \leq bax \leq ba)$ and select $C_j$ from the ODM by using the $bax$ and recent city visited by the individual. If

$C_j$ is not visited by the individual then visit $C_j$. If all the best adjacent cities are visited by the individual, then randomly generate $C_j$ where $C_j \notin Ind_i$ then visit $C_j$. This process continues until all the individuals for the population (POP) are generated. Then select the leader individual from the population where leader individuals of set $IndL$ are having the highest fitness values in population and the remaining individuals from the population are active individuals of set $IndA$.

Sequences are discovered from the leader individuals by comparing the two leader individual $Ind_i$ and $Ind_j$ from the set $IndL$. The sequence $seq_{r,s}$ is the set of cities visited in same order by the individual $Ind_i$ and $Ind_j$. The sequences are of length are formed with sub-tour using the Equation (10). If the city matches then repeat the same step until the sequence $(seq_{r,s})$ of length $s$ is formed. Repeat the above process from selecting next two individual until all the individual in leader individuals are compared and find all the possible $Seq$.

Active individuals $(IndA)$ are having the low fitness values when compared with the leader individuals $(IndL)$. In order to increase the fitness value of active individual $(IndA)$, active individuals are incorporated with those sequences $(seq_{r,s})$ followed by the leaders individuals. Then $IndA$ reduce the distance by visiting all the cities and increase the fitness value of $IndA$ and move towards $IndL$. In the common pattern replacement model, the replacement process contains the whole higher values of '$v$' that lead to struck with local optima. The replacement process in terms of sub patterns rather than the whole pattern generates good results by minimizing the probability of trapping with local optima. It can be proved by Empirical analyses that the subpatterns with the length of 4 yields better results. Select the sequence from the set $seq_{s,r}$ and check whether $Seq_{1,r}$ have some matching sequence $MSeq$ in active individuals.

Then the Greedy Crossover technique, which sometimes referred as Greedyswap is used here. The basic idea of Greedy Crossover is to randomly select two cities from one Individual and swap them if the new (swapped) tour length is shorter than the old one.

This leads to an always-converging solution pointing towards the shortest tour length of a *TSP*. The comparison is made prior to crossover, which brings to the concluding factor that only new individual with high fitness values are created. After the crossover, the individuals are subjected to mutation. The swap mutation operator, which is commonly used for permutation based representation is applied here. It simply consists of generating a new individual by randomly swapping two cities from an existing one. The mutation probability $Pm$ is usually taken as $1/N$.

## 5. Experimental Design

### 5.1. Experimental Setup

The competence of the proposed self-organization model for the *GA* is demonstrated using a set of well-defined experiments. Traveling Salesman Problem (*TSP*), a popular combinatorial optimization *NP* hard problem is being chosen as the testbed. The standard benchmark *TSP* instances obtained from *TSPLIB* [33], named as Eil51, Pr76, KroA100, Pr144, Gil262, Fl417 and u724 are considered for experiments reported in this paper.

The individuals of the population are represented as the tours of cities. Experiments are conducted in three different phases: Phase-1 is used to assess the performance of the proposed self-organization model (SOR) in conjunction with the Random (*R*) population seeding technique [15, 28, 29] in a classical GA; Phase-2 is used to assess the performance of the proposed self-organization model (SONN) in conjunction with the Nearest Neighbor (*NN*) population seeding technique [31, 38] in a classical GA; Phase-3 is used to assess the performance of the proposed self-organization model (SOVV) in conjunction with the *VV* population seeding technique [34, 49] in a classical GA. For all the phases of experiments, the no. of individuals are defined as 100 and the total no. of generations are limited as 100. For each case of assessment, the average of the 50 independent runs is considered with the crossover probability as 0.7 $(Pc = 0.7)$ and the mutation probability as 0.1 $(Pm = 0.1)$. All the techniques are assessed using a set of generic performance criteria, which are described as in the next section.

### 5.2. Performance Assessment Criteria

The parameters used for the investigation of the proposed technique are defined such that to assess the performance of a classical *GA* in general and so, it could explore the abilities of the same in various dimensions along with the trade-offs. The assessment criteria used in this investigation are convergence rate, error rate, average convergence and distinct individuals from the population, and *NN* ratio. The Nearest Neighbor (*NN*) ratio of the population is the ratio of the total number of nearest neighbor edges to the total number of edges in individuals of the population, where the nearest neighbor edge is an edge at least one of its end cities is the nearest neighbor of the other end city [33].

Convergence rate of an individual of a population set is defined as the percentage of fitness value obtained by the individual according to the optimal fitness value and it can be defined as in the Equation (12) as follows [29, 33, 34, 49, 54, 56]:

$$\text{con. rate } (\%) = 1 - \frac{\text{Fitness } - \text{optimal fitness}}{\text{optimal fitness}} \times 100. \tag{12}$$

Average convergence rate of a population set is defined as the average percentage of fitness value obtained by the individual according to the optimal fitness value and it can be defined as in the Equation (13) as follows [10, 33, 49, 56]:

$$\text{Ave. converg.}(\%) = 1 - \frac{\text{average fitness } - \text{optimal fitness}}{\text{optimal fitness}} \times 100. \tag{13}$$

Error rate of an individual of a population set is defined as the percentage of difference between fitness obtained by the individual and optimal fitness value [10, 39, 41]. It can be defined as follows:

$$\text{Errorrate } (\%) = \frac{\text{Fitness } - \text{optimal fitness}}{\text{optimal fitness}} \times 100. \tag{14}$$

Distinct individual are the different individuals from the population. It is also an important factor that the increase and decrease in distinct individual reduces the performance [42].

**Table 1.** Random Vs Self-Organized Random Technique

| Sl.No. | TSP Instance | Initial. Tech. | Comp. Time | Opt. Value | Fitness | | | Convergence rate (%) | | Error rate (%) | | Ave. Converg. (%) | NN Ratio (%) | Distinct Ind. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Best | Worst | Average | Best | Worst | Best | Worst | | | |
| 1 | eil51 | R | 43.49 | 426 | 522.44 | 1022.53 | 890.66 | 77.46 | -40.03 | 22.64 | 140.03 | -9.08 | 16.08 | 76 |
| | | SOR | 271.98 | | 491.69 | 926.80 | 819.14 | 84.58 | -17.56 | 15.42 | 117.56 | 7.71 | 36.57 | 85 |
| 2 | pr76 | R | 70.39 | 108159 | 131082.80 | 239676.20 | 218945.99 | 78.81 | -21.60 | 21.19 | 121.60 | -2.43 | 7.64 | 82 |
| | | SOR | 450.36 | | 125000.88 | 217892.66 | 201945.99 | 84.43 | -1.46 | 15.57 | 101.46 | 13.29 | 29.45 | 71 |
| 3 | rat99 | R | 92.42 | 1211 | 1480.14 | 2897.43 | 2590.06 | 77.78 | -39.26 | 22.22 | 139.26 | -13.88 | 5.65 | 88 |
| | | SOR | 362.98 | | 1374.82 | 2568.18 | 2362.00 | 86.47 | -12.07 | 13.53 | 112.07 | 4.95 | 26.87 | 74 |
| 4 | kroA100 | R | 93.56 | 21282 | 27883.73 | 52237.00 | 44651.89 | 68.98 | -45.45 | 31.02 | 145.45 | -9.81 | 3.72 | 78 |
| | | SOR | 1245.38 | | 25136.36 | 45904.20 | 41951.52 | 81.89 | -15.69 | 18.11 | 115.69 | 2.88 | 27.56 | 72 |
| 5 | pr144 | R | 141.12 | 58537 | 71188.74 | 149016.52 | 125840.30 | 78.39 | -54.57 | 34.64 | 154.57 | -14.98 | 2.22 | 85 |
| | | SOR | 734.12 | | 67893.29 | 130721.07 | 115544.85 | 84.02 | -23.31 | 15.98 | 123.31 | 2.61 | 23.77 | 79 |
| 6 | tsp225 | R | 229.07 | 3919 | 4803.16 | 10361.91 | 8953.00 | 77.44 | -64.40 | 22.56 | 164.40 | -28.45 | 1.05 | 81 |
| | | SOR | 2940.15 | | 4469.76 | 9404.06 | 7934.61 | 85.95 | -39.96 | 14.05 | 139.96 | -2.47 | 18.18 | 74 |
| 7 | fl417 | R | 483.45 | 11861 | 14945.57 | 34181.60 | 27387.99 | 73.99 | -88.18 | 26.01 | 188.18 | -30.91 | 0.81 | 92 |
| | | SOR | 4283.69 | | 14152.41 | 29213.51 | 24674.64 | 80.68 | -46.30 | 19.32 | 146.30 | -8.03 | 10.73 | 73 |
| 8 | u724 | R | 944.55 | 41910 | 54068.09 | 121805.62 | 99013.84 | 70.99 | -90.64 | 29.01 | 190.64 | -36.25 | 0.31 | 87 |
| | | SOR | 7856.89 | | 52017.97 | 108484.53 | 88617.09 | 75.88 | -58.85 | 24.12 | 158.85 | -11.45 | 4.27 | 78 |

Init. Tech. = Initialization Technique; Comp. Time = Computation Time;
Opt. Value = Optimum Value; Ave. Converg. = Average Convergence;
Distinct Ind. = Distinct Individual.

**Table 2.** Nearest Neighbour Vs Self-Organized Nearest Neighbour Technique

| Sl.No. | TSP Instance | Initial. Tech. | Comp. Time | Opt. Value | Fitness | | | Convergence rate (%) | | Error rate (%) | | Ave. Converg. (%) | NN Ratio (%) | Distinct Ind. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Best | Worst | Average | Best | Worst | Best | Worst | | | |
| 1 | eil51 | NN | 33.46 | 426 | 489.09 | 685.78 | 626.44 | 85.19 | 39.02 | 14.81 | 60.98 | 52.95 | 64.16 | 39 |
| | | SONN | 189.14 | | 463.49 | 664.18 | 588.24 | 91.20 | 44.09 | 8.80 | 55.91 | 61.92 | 49.60 | 54 |
| 2 | pr76 | NN | 61.01 | 108159 | 122000.59 | 170440.39 | 165084.50 | 87.20 | 42.42 | 12.80 | 57.58 | 47.37 | 59.45 | 45 |
| | | SONN | 373.36 | | 112725.77 | 160463.83 | 157399.75 | 95.78 | 51.64 | 4.22 | 48.36 | 54.47 | 52.15 | 61 |
| 3 | rat99 | NN | 67.03 | 1211 | 1351.05 | 2065.92 | 1910.60 | 88.44 | 29.40 | 11.56 | 70.60 | 42.23 | 62.82 | 34 |
| | | SONN | 292.98 | | 1260.30 | 2031.35 | 1824.40 | 95.93 | 32.26 | 4.07 | 67.74 | 49.35 | 51.54 | 53 |
| 4 | kroA100 | NN | 70.40 | 21282 | 23727.76 | 36985.82 | 34333.56 | 88.51 | 26.21 | 11.49 | 73.79 | 38.67 | 62.68 | 37 |
| | | SONN | 1174.38 | | 23620.08 | 36030.22 | 33792.28 | 89.01 | 30.70 | 10.99 | 69.30 | 41.22 | 55.46 | 48 |
| 5 | pr144 | NN | 94.08 | 58537 | 64808.37 | 104615.59 | 90970.95 | 89.29 | 21.28 | 10.71 | 78.72 | 44.59 | 76.04 | 34 |
| | | SONN | 896.46 | | 60885.71 | 102059.99 | 84803.21 | 95.99 | 25.65 | 4.01 | 74.35 | 55.13 | 63.00 | 55 |
| 6 | tsp225 | NN | 160.65 | 3919 | 4619.35 | 6864.59 | 6296.46 | 82.13 | 24.84 | 17.87 | 75.16 | 39.33 | 68.09 | 39 |
| | | SONN | 2376.88 | | 4228.56 | 6332.86 | 5850.03 | 92.10 | 38.41 | 7.90 | 61.59 | 50.73 | 53.36 | 52 |
| 7 | fl417 | NN | 339.84 | 11861 | 14597.00 | 21721.28 | 19385.66 | 76.93 | 16.87 | 23.07 | 83.13 | 36.56 | 63.75 | 34 |
| | | SONN | 4285.55 | | 13144.25 | 21207.95 | 18728.36 | 89.18 | 21.20 | 10.82 | 78.80 | 42.10 | 50.61 | 56 |
| 8 | u724 | NN | 593.16 | 41910 | 51210.90 | 72857.82 | 70221.87 | 77.81 | 26.16 | 22.19 | 73.84 | 32.45 | 66.66 | 33 |
| | | SONN | 7378.12 | | 46355.70 | 69906.22 | 63277.27 | 89.39 | 33.20 | 10.61 | 66.80 | 49.02 | 49.32 | 49 |

Init. Tech. = Initialization Technique; Comp. Time = Computation Time;
Opt. Value = Optimum Value; Ave. Converg. = Average Convergence;
Distinct Ind. = Distinct Individual.

**Table 3.** Vv Vs Self-Organized Vv Technique

| Sl.No. | TSP Instance | Initial. Tech. | Comp. Time | Opt. Value | Fitness | | | Convergence rate (%) | | Error rate (%) | | Ave. Converg. (%) | NN Ratio (%) | Distinct Ind. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Best | Worst | Average | Best | Worst | Best | Worst | | | |
| 1 | eil51 | VV | 46.27 | 426 | 453.70 | 585.78 | 507.44 | 93.50 | 62.49 | 6.50 | 37.51 | 80.88 | 47.33 | 74 |
| | | SOVV | 234.98 | | 432.10 | 550.18 | 485.84 | 98.57 | 70.85 | 1.43 | 29.15 | 85.95 | 53.81 | 68 |
| 2 | pr76 | VV | 65.21 | 108159 | 113657.75 | 160440.39 | 128063.45 | 94.92 | 51.66 | 5.08 | 48.34 | 81.60 | 53.67 | 48 |
| | | SOVV | 416.74 | | 109723.70 | 140463.83 | 124514.83 | 98.55 | 70.13 | 1.45 | 29.87 | 84.88 | 49.40 | 58 |
| 3 | rat99 | VV | 89.37 | 1211 | 1317.56 | 1765.92 | 1523.41 | 91.20 | 54.18 | 8.80 | 45.82 | 74.20 | 52.60 | 51 |
| | | SOVV | 320.31 | | 1241.75 | 1531.35 | 1401.81 | 97.46 | 73.55 | 2.54 | 26.45 | 84.24 | 55.96 | 61 |
| 4 | kroA100 | VV | 90.56 | 21282 | 22539.64 | 29985.82 | 25973.40 | 94.09 | 59.10 | 5.91 | 40.90 | 77.96 | 52.20 | 50 |
| | | SOVV | 1282.96 | | 21584.04 | 27030.22 | 25017.80 | 98.58 | 72.99 | 1.42 | 27.01 | 82.45 | 56.23 | 59 |
| 5 | pr144 | VV | 123.56 | 58537 | 61663.21 | 84615.59 | 72425.48 | 94.66 | 55.45 | 5.34 | 44.55 | 76.27 | 62.01 | 55 |
| | | SOVV | 694.12 | | 59707.61 | 75059.99 | 70469.88 | 98.00 | 71.77 | 2.00 | 28.23 | 79.61 | 65.43 | 60 |
| 6 | tsp225 | VV | 205.45 | 3919 | 4610.68 | 5864.59 | 4905.71 | 82.35 | 50.36 | 17.65 | 49.64 | 74.82 | 51.71 | 60 |
| | | SOVV | 2423.88 | | 4061.27 | 5432.86 | 4467.32 | 96.37 | 61.37 | 3.63 | 38.63 | 86.01 | 64.00 | 64 |
| 7 | fl417 | VV | 411.89 | 11861 | 13181.50 | 17721.28 | 15378.39 | 88.87 | 50.59 | 11.13 | 49.41 | 70.34 | 45.61 | 61 |
| | | SOVV | 4862.53 | | 12668.17 | 16207.95 | 14865.06 | 93.19 | 63.35 | 6.81 | 36.65 | 74.67 | 41.63 | 64 |
| 8 | u724 | VV | 748.23 | 41910 | 44907.52 | 60857.82 | 49223.53 | 92.85 | 54.79 | 7.15 | 45.21 | 82.55 | 50.78 | 60 |
| | | SOVV | 7816.89 | | 43955.92 | 56906.22 | 48271.93 | 95.12 | 64.22 | 4.88 | 35.78 | 84.82 | 53.09 | 62 |

Init. Tech. = Initialization Technique; Comp. Time = Computation Time; Opt. Value = Optimum Value; Ave. Converg. = Average Convergence; Distinct Ind = Distinct Individual.

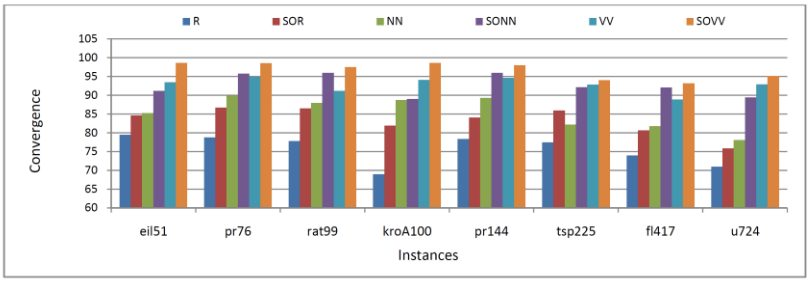



**Figure 3.** Performance Analyses in terms of Best Convergence Rate.


**Figure 4.** Performance Analyses in terms of Worst Convergence Rate.

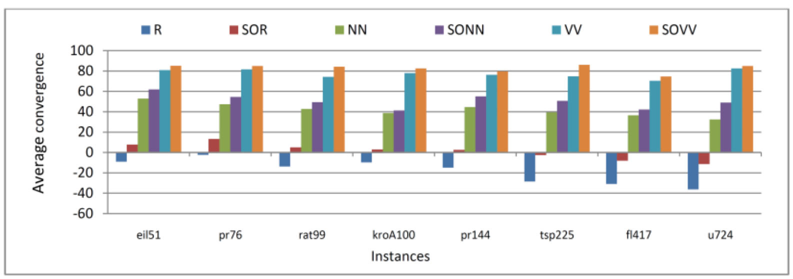



**Figure 5.** Performance Analyses in terms of Average Convergence Rate.

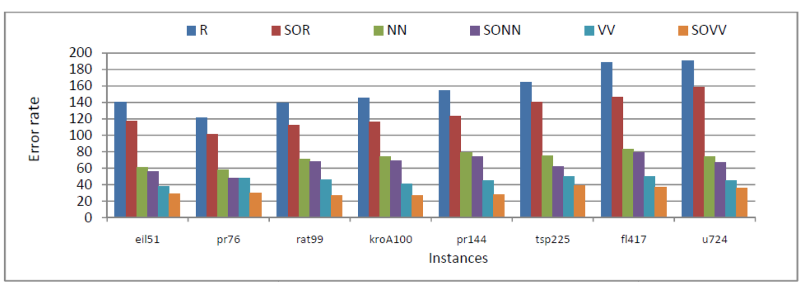



**Figure 6.** Performance Analyses in terms of Error Rate.

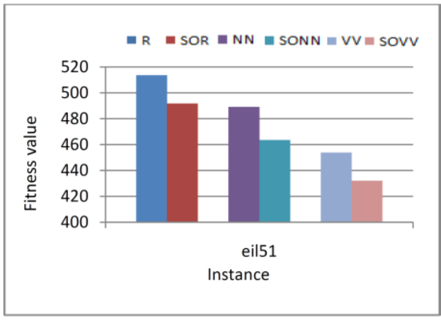



**Figure 7.** Fitness value w.r.t. eil51 Instance

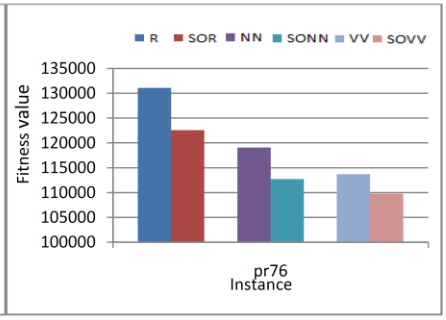



**Figure 8.** Fitness value w.r.t. pr76 Instance

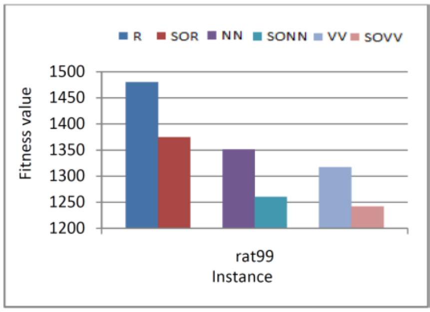



**Figure 9.** Fitness value w.r.t. rat99 Instance

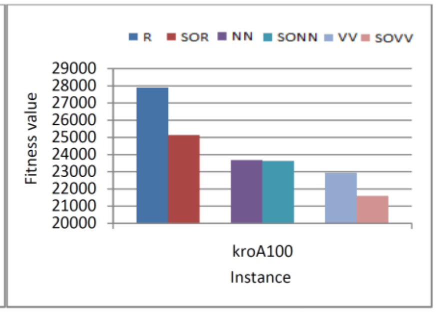



**Figure 10.** Fitness value w.r.t. kro100 Instance

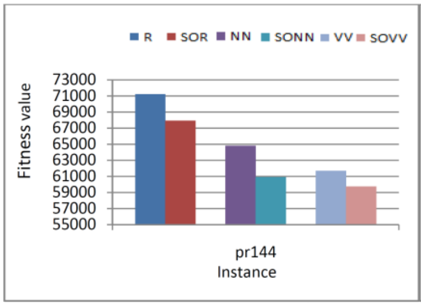



**Figure 11.** Fitness value w.r.t. pr144 Instance

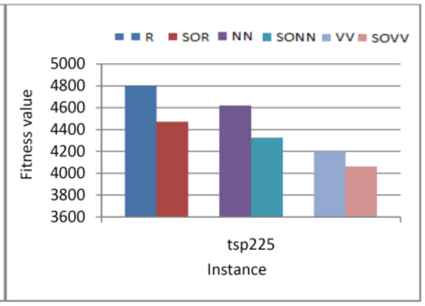
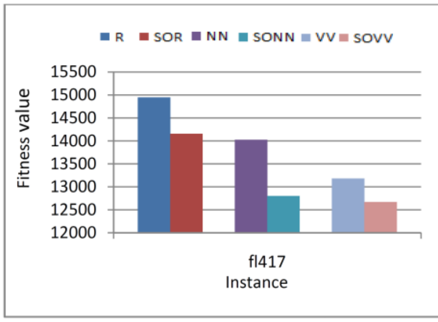



**Figure 12.** Fitness value w.r.t. tsp225 Instance



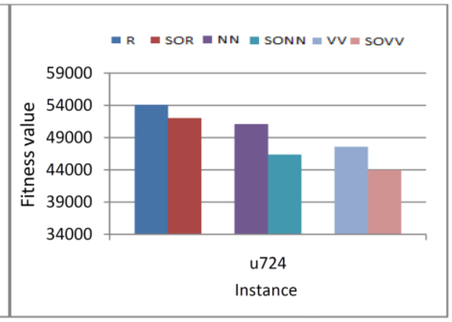




**Figure 13.** Fitness value w.r.t. fl417 Instance

**Figure 14.** Fitness value w.r.t. u724 Instance

**Figure 7-14.** Performance Analyses in terms of Fitness value w.r.t. TSPLIB Instances.

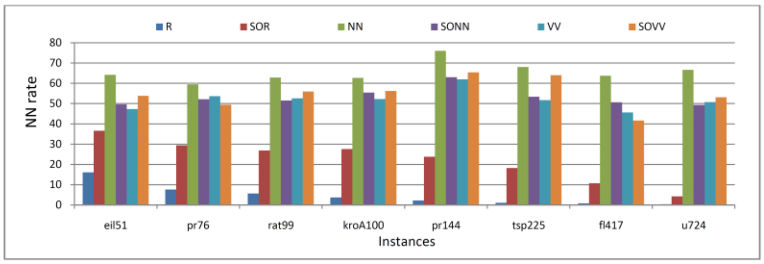



**Figure 15.** Performance Analyses in terms of Nearest Neighbor Rate.

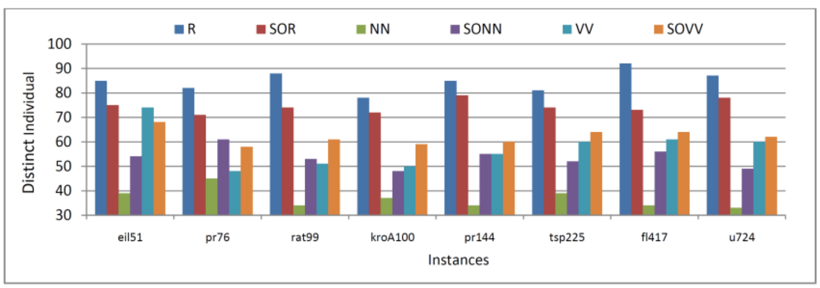



**Figure 16.** Performance Analyses in terms of Distinct Individual.

## 6. Experiments and Result Analyses

As described earlier, the experiments of this investigation are conducted in three different phases in terms of the existing initialization techniques and proposed enhancements for the same set.

### 6.1. Phase-I: Random (R) Vs. Self-Organized Random (SOR) Technique

In this phase, the random technique [15, 28, 29, 30] and its enhanced model are being used as the initialization technique for initial population generation in a classical *GA*. The city visits are formulated randomly without any heuristics. Table I depicts the outcomes and observations of this phase of experiments. i.e. performance of the random technique (*R*) and its enhancement model as self-organized random (SOR) technique. It is observed that, it terms of convergence rate, the proposed SOR technique dominates the random technique. For sample instance eil51, in case of random technique, the best convergence rate obtained is around 77.46%, whereas the same for the proposed SOR technique is as 84.58%. Similarly, the average convergence value of the former one for the same instance is around -9.08%, whereas the SOR technique achieved the rate of 7.71% for the same. On the other hand, in terms of error rate also, this form of supremacy is continued for all instances. For example, for the same sample instance eil51, the random technique obtained the best error rate as 22.64% and the worst rate as 140.03%. But, the SOR technique obtained the same in the order of 15.42% and 117.56% respectively for the same instance.

Likewise, the outcomes in terms of *NN* ratio of the proposed SOR technique lead its existing version. For the sample case, the experiments corresponds to the instance eil51 yield the average value of *NN* ratio as 16.08% while its enhanced model yielded the respective value as 36.57%, which is almost twice of the former one. In the same way, in terms of fitness value also the proposed model outperforms its existing version. For instance, the average fitness value of the instance eil51 is achieved as 890.66 by the random technique, where the SOR offered the same as 819.14, which is definitely comparatively better than the former one. In case of the distinct individual, the later technique produces an average of 85 distinct individuals as compared with the former one, which produced 76 distinct individuals that shows the leading performance of the proposed technique. This large range of distinct individuals proved the broader exploring capability of the proposed technique against its existing version. This kind of ascendancy is continued for all the instances in terms of all the criteria as discussed above.

### 6.2. Nearest Neighbor (NN) Vs. Self-Organized Nearest Neighbor (SONN) Technique

Nearest Neighbor (*NN*) technique [31, 38] and its self-organized version SONN are used for initial population generation in a classical GA in this phase. Here the selection of cities to be visited is made as function of distance between the same. Preference is given to the cities with minimum distance values and it is made by a set of heuristics.

The experiments are conducted over the instances as described in the experimental set-up and as same as in the Phase-I. The corresponding outcomes of these experiments are recorded in the Table II. In this case also, the proposed SONN technique lead over the *NN* technique by all means. For example, in case of the instance pr76, the best convergence rate of the existing *NN* technique is around 87.20%, whereas the SONN technique produced the same as 95.78%. Equally, the average convergence rate of the NN technique is about 47.37%, while the other has the same as 54.47%, which clearly illustrate the prevailing performance of the proposed SONN technique. In addition to that, the error rate of the proposed technique is low as compared with the existing technique. By considering the same instance pr76, it is observed that the best error rate of the proposed technique is only around 4.22% and the worst rate is around 48.36%, however the same of the existing technique is about 12.80% and 57.58% respectively.

Correspondingly, in terms of fitness value also, the new technique of SONN leads the existing *NN* technique. For discussion, the SONN technique obtains the average fitness value as 157399.75 for the instance pr76, but the NN technique achieved the same as 165084.50. This clearly explored the competency of the proposed technique against the existing version. Respectively, the exploring ability of proposed technique is also proved as quite high compared to the existing set. The new technique produced the average of 61 distinct individuals; however the existing one explores only 45 distinct individuals, which visibly demonstrated the narrowed exploration ability of the existing model. On contrary to the above discussions, the proposed technique lags in terms of *NN* ratio w.r.t. the existing model. In detail, in case of the instance pr76, it is observed that the SONN technique accomplished the *NN* ratio as 52.15%, nevertheless the existing one has the ability of around 59.45%. This stream of supremacy is observed across all the instances of this phase of experiments in terms of the criteria as discussed respectively.

**6.3. VV Vs Self-Organized VV Technique**

This phase of experiments has used the VV technique [34, 49] and its variant SOVV as initialization techniques for a classical GA in this phase and so for performance assessments. Here the cities are visited based on heuristics as a composite function of randomness and distance. As described in the experimental design section, experiments are conducted over the TSP instances as in the Phase-I and the corresponding results are presented in the Table III. As like in the previous experimental phases, the proposed SOVV technique presides over the existing VV technique. For the detailed discussion, by considering the sample instance rat99, it is observed that the former one is leading the later one by all convergence arte as 73.55%, where as the existing VV technique accomplished the best convergence rate as 91.20% and the worst convergence rate as 54.18% respectively. Similarly, the SOVV technique produced the average convergence rate as 84.24%; conversely the VV technique produced the same as 74.20% only. In summary, it is evidently proved that the proposed technique outperforms the existing technique in terms of convergence by all means. As an accumulative indication, the achieved error rate of the SOVV technique for the same instance rat99 is very less as the best error rate as 2.54% and the worst rate is around 26.45%; on the other hand the best error rate of the existing technique is about 12.80% and 57.58% respectively. It proves the effectiveness of the proposed technique in comparison with its existing version.

In the same way, the SOVV technique produced leading performance in terms of fitness value over the classical VV technique. For example, for the instance rat99, the average fitness value achieved by the SOVV technique as 1401.81, while the other reached the average fitness value as 1523.41. This detailed the preeminence of the proposed technique against the existing one. In the same order, the exploring ability of proposed technique is also proved in terms of the no. of distinct individuals: the SOVV technique produced the average of 61 distinct individuals, whereas the existing one explores only 51 distinct individuals, which obviously demonstrated the exploration capability of the proposed model. In addition to these, it is also observed that the existing technique lags in terms of NN ratio (whose value is 52.60%) w.r.t. the SOVV model (whose value is 55.96%). These analyses are undoubtedly demonstrated the ability of the proposed technique in all aspects.

### 6.4. Discussion

In all the three phases of experiments, the abilities of the proposed self-organized models are demonstrated in terms of the well-defined and globally accepted performance factors. In all the cases, it is proved that the proposed self-organized models of the respective population seeding techniques are outperformed when compared with their existing classical models. In detail, in terms of convergence rate, by all means, the proposed self-organizing models perform better than their classical versions. The overall performance in terms of the best convergence rate is clearly illustrated in the Figure 4 and in terms of worst convergence rate is depicted in the Figure 5. In this sequence, the overall performance comparison in terms of average convergence rate is shown in the Figure 6. Similarly, in terms of error rate also, the proposed self-organizing models lead their classical versions in all aspects. This dominant performance is visibly exemplified as in the Figure 7, which shows the leading performance of the proposed models in comparison with the existing ones in terms of error rates respectively. In terms of fitness value also, the proposed models perform better than their counterparts and it is noticeably represented in the Figures 8 to 15. Likewise, the proposed models outperform the existing ones w.r.t. NN rate and distinct individuals and the same are characterized in the Figures 16 and 17 respectively. In this case, the illustrations are made w.r.t. the individual TSPLIB instances and it is proved that respective enhancement models presented better performance rather than their classical versions. In summary, it is experimentally proved that in combination with the proposed self-organizing models, the existing classical Genetic Algorithms work better in most of the situations.

## 7. Conclusion

This work reported in this paper proposed a common pattern replacement model based self-organization technique, which is combined with the classical GA in order to enhance the overall performance of the same. The significance of the proposed self-organization technique for the *GA* is demonstrated using well-defined experiments conducted over the Traveling Salesman Problem

(TSP). The standard benchmark TSP instances obtained from TSPLIB [33], named as Eil51, Pr76, KroA100, Pr144, Gil262, Fl417 and u724 are considered for experiments reported in this paper. The individuals of the population are represented as the tours of cities. Experiments are conducted in three different phases and for all the phases of experiments, the no. of individuals are defined as 100 and the total no. of generations are limited as 100. For each case of assessment, the average of the 50 independent runs is considered with the crossover probability as 0.7 $(Pc = 0.7)$ and the mutation probability as 0.1 $(Pm = 0.1)$. In all the three phases of experiments, the abilities of the proposed self-organized models are demonstrated in terms of the well-defined and globally accepted performance factors. In all the cases, it is proved that the proposed self-organized models of the respective population seeding techniques are outperformed when compared with their existing classical models. The outcomes of the experiments evidently proved that the proposed self-organization technique in combination with the classical *GA* avoids the premature convergence and improves the error and diversity rates by all means. These outcomes also encouraged to continue this research such that to enhance the proposed model for the large scale optimization problems.

## Acknowledgements

This work is a part of the Research Projects sponsored by Visvesvaraya Ph.D. Scheme for Electronics & IT, Ministry of Electronics & Information Technology, India, and Reference No: PHD-MLA-4(44)/2015-16, dated August 2015. The authors would like to express their thanks for the financial supports offered by the Sponsored Agency.

## References

[1] S. Abe, Convergence acceleration of the Hopfield neural network by optimization integration step sizes, IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 26(1) (1996), 194-201.

[2] Chang Wook Ahn, Ki Pyo Kim and R. S. Ramakrishna, A memory-efficient elitist genetic algorithm, Parallel Processing and Applied Mathematics 3019 (2004), 552-559.

[3] H. Al-Mulhem and T. Al-Maghrabi, Efficient convex-elastic net algorithm to solve the Euclidean travelingsalesmanproblem, IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 28(4) (1998), 618-20.

[4] J. Amudhavel, U. Prabu, N. Saravanan, P .Dhavachelvan, R. Baskaran and V. S. K. Venkatachalapathy, A Distributed Spanning Tree Based Dynamic Self Organizational framework for Web Server, Proceedings of Second International Conference on Computer and Communication Technologies, ISBN: 978-81-322-2525-6, 3, Advances in Intelligent Systems and Computing, ISSN: 2194-5357, 381 (2015), 333-339, July 24th-26th, Hyderabad, INDIA, 2015.

[5] J. Amudhavel, S. Padmapriya, R. Nandhini, G. Kavipriya, P. Dhavachelvan and V. S. K. Venkatachalapathy, Recursive Ant Colony Optimization Routing in wireless mesh network, Springerlink AISC Indexed International Conference on Computer and Communication Technologies, ISBN: 978-81-322-2525-6, Volume 3, Advances in Intelligent Systems and Computing, ISSN: 2194-5357, 381 (2015), 341-351, July 24th-26th, Hyderabad, INDIA, 2015.

[6] Bernard Angeniol, Gael de La Croix Vaubois and Jean-Yves Le Texier, Self-organizing feature maps and the traveling salesman problem, Neural Networks 1(4) (1988), 289-293.

[7] N. Aras, B. J. Oommen and I. K. Altinel, Kohonen Network incorporating explicit statistics and its application to the traveling salesman problem, Neural Networks 12(9) (1998), 1273 -1284.

[8] J. Amudhavel, N. Moganarangan, P. Dhavachelvan, R. Baskaran, M. S. Saleem Basha, A. K. Rehana Begum and K. Prem Kumar, Krill Herd Optimization based Fault Tolerance strategy in MANETS for Dynamic Mobility, Proceedings of Indexed International Conference on Circuit, Power and Computing Technologies, ISBN: 978-1-4799-7074-2, pp. 1-7, March 19th-20th, Kanyakumari, Tamilnadu, INDIA, 2015.

[9] Bala and A. K. Sharma, A Comparative Study of Modified Crossover Operators (2015), 281-284.

[10] R. J. Balling, R. R. Briggs and K. Gillman, Multiple optimum size/shape/topology designs for skeletal structures using a genetic algorithm, Journal of Structural Engineering 132(7) (2006), 1158-1165. ISSN: 1943-541X.

[11] Bak and Per, How Nature Works: The Science of Self-Organized Criticality, Copernicus Press, New York, XIII, (1996), 212 p.

[12] Heni Ben Amor and Heni Ben Amor, Intelligent Exploration for Genetic Algorithms Using Self Organizing Maps in Evolutionary Computation, GECCO'05, June 25-29, Washington, DC, (2005), 1531-1538.

[13] V. Boyer, D. El Baz and M. Elkihel, A dynamic programming method with dominance technique for the knapsack sharing problem, Int. Conf. Comput. Ind. Eng. (2009), 348-353.

[14] M. Budinich, A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing, Neural Computation 8(2) (1996), 416-442.

[15] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, & Eric Bonabeau, Self-Organizing in Biological Systems, Princeton University Press: New Jersey, 560 pp.

[16] M. Chen, C. Gong, X. Li, Z. Yu and A. G. Process, Based on Virtual Instrument Technology and Genetic-Annealing Algorithms (2015) 1825-1827.

[17] Kusum Deep and Dipti, A self-organizing migrating genetic algorithm for constrained optimization, Applied Mathematics and Computation 198(1) (2008), 237-250.

[18] S. Favata and R. Walker, A study of the application of Kohonen-type neural networks to the traveling salesman problem, Biological Cybernetics 64(6) (1991), 463-468.

[19] J. C. Fort, Solving combinatorial problem via self-organizing process: an application of the Kohonen algorithm to the traveling sales-man problem, Biological Cybernetics 59(1) (1988), 33-40.

[20] Dimitris Fotakis and Epameinondas Sidiropoulos, A new multi-objective self-organizing optimization algorithm (MOSOA) for spatial optimization problems, Applied Mathematics and Computation 218(9) (2012), 5168-5180.

[21] John H. Holland, Adaptation in Natural and Artificial Systems, MIT Press Cambridge, MA, USA, (1992).

[22] G. J. Hueter, Solution of the traveling salesman problem with an adaptive ring, IEEE International Conference on Neural Networks, (1988).

[23] Kwon Jeong and Jang Lee, A self-organizing genetic algorithm for multimodal function optimizer, Artif Life and Robotics 2(1) (1998), 48-52.

[24] Zhang Jinhua, Zhuang Jian, Du Haifeng and Wang Sun'an, Self-organizing genetic algorithm based tuning of PID controllers, Information Sciences, 179(7) (2009), 1007-1018.

[25] Eisuke Kita, Shen Kan and Zhai Fei, Investigation of self-organizing map for genetic algorithm, Advances in Engineering Software 41(2) (2010), 148-153.

[26] Rae-Goo Kang and Chai-Yeoung Jung, The Improved Initialization Method of Genetic Algorithm for Solving the Optimization Problem, Neural Information Processing 4234 (2006), 789-796.

[27] T. Kohonen, M. R. Schroeder and T. S. Huang, Self-Organizing Maps, Springer-Verlag New York, Inc. Secaucus, NJ, USA, (2001).

[28] Leila Kallel and Marc Schoenauer, Alternative Random Initialization in genetic algorithm, Proceedings of the 7$^{th}$ International Conference on Genetic Algorithms, (1997).

[29] K. Katayama, H. Sakamoto and H. Narihisa, The efficiency of hybrid mutation Genetic Algorithm for the Travelling Salesman Problem, Mathematical and Computer Modeling, Elsevier, Issue 31, October, (2000), 197-203.

[30] J. S. Lawrence and M. M. Amini, Performance characteristics of alternative genetic algorithmic approaches to the traveling salesman problem using path representation: an empirical study, European Journal of Operational Research (1998), 551-570.

[31] Y. Lu, S. Lu, F. Fotouhi, Y. Deng and S. Brown, (2004), Incremental genetic $K$-means algorithm and its application in Gene Expression Data Analysis, BMC Bioinformatics 5 (1996), 172.

[32] Y. Mei, X. Li, F. Salim and X. Yao, Heuristic evolution with Genetic Programming for Traveling Thief Problem, IEEE Congr. Evol. Comput. CEC 2015 - Proc. (2015), 2753-2760.

[33] Hiroyuki Okano, Shinji Misono and Kazuo Iwano, New TSP Construction Heuristics and Their Relationships to the 2-Opt, Journal of Heuristics 5(1) (1999), 71-88.

[34] P. Victer Paul, A. Ramalingam, R. Baskaran, P. Dhavachelvan, K. Vivekanandan and R. Subramanian, A new population seeding technique for permutation coded Genetic Algorithm: Service transfer approach, Journal of Computational Science, Elsevier 5(2) (2013), 277-297.

[35] D. T. Pham and T. T. B. Huynh, An effective combination of Genetic Algorithms and the Variable Neighborhood Search for solving Travelling Salesman Problem, TAAI. Technol. Appl. Artif. Intell. (2015), 142-149.

[36] R. Raju, J. Amudhavel, N. Kannan and M. Monisha, A Bio inspired Energy-Aware Multi objective Chiropteran Algorithm (EAMOCA) for hybrid cloud computing environment, Proceedings of International Conference on Green Computing Communication and Electrical Engineering, ISBN: 978-1-4799-4982-3, pp.1-5, March 6th-8th, Coimbatore, INDIA, 2014.

[37] R. Raju, J. Amudhavel, N. Kannan and M. Monisha, Interpretation and evaluation of various hybrid energy aware technologies in cloud computing environment-A detailed survey, Proceedings of International Conference on Green Computing Communication and Electrical Engineering, ISBN: 978-1-4799-4982-3, 1-3, March 6th-8th, Coimbatore, INDIA, 2014.

[38] Shubhra Sankar Ray, Sanghamitra Bandyopadhyay and Sankar K. Pal, Genetic operators for combinatorial optimization in TSP and microarray gene ordering, Applied Intelligence 26(3) (2007), 183-195.

[39] Kwong-SakLeunga, Hui-Dong Jinb and Zong-Ben Xuc, An expanding self-organizing neural network for the traveling salesman problem, Neurocomputing 62 (2004), 267-292.

[40] T. D. Seeley, When Is Self-Organizing Used in Biological Systems?, Biol. Bull. 202 (2002), 314-318.

[41] S. Somhom, A. Modares and T. Enkawa, Competition-based neural network for the multiple traveling salesman problem with minmax objective, Computer Res. (1999), 395-407.

[42] Sonmez, E. Kocyigit and E. Kugu, Optimal path planning for UAVs using Genetic Algorithm, Int. Conf. Unmanned Aircr. Syst. (2015), 50-55.

[43] G. Singh, New crossover operators for Real Coded Genetic Algorithm (RCGA ), (2015), 135-140.

[44] Renato Tinos and Shengxiang Yang, Genetic Algorithms with Self-Organizing Behaviour in Dynamic Environments, Studies in Computational Intelligence 51 (2007), 105-127.

[45] V. Togan and A. T. Daloglu, An improved genetic algorithm with initial population strategy and self-adaptive member grouping, Journal of Computers and Structures (2008), 1204-1218.

[46] Renato Tinos and Shengxiang Yang, A Self-Organizing Random Immigrants Genetic Algorithm for Dynamic Optimization Problems, Genetic Programming and Evolvable Machines 8(3) (2007), 255-286.

[47] V. Togan and A. T. Daloglu, Optimization of 3d trusses with adaptive approach in genetic algorithms, Engineering Structures 28(7) (2006), 1019-1027.

[48] V. Togan, A. T. Daloglu, Adaptive approaches in genetic algorithms to catch the global optimum, Proc 7th international congress advances in civil engineering, Istanbul, (2006).

[49] P. Victer Paul, R. Baskaran and P. Dhavachelvan, A Novel Population Initialization Technique for Genetic Algorithm, IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT), India, (2013), 1235-1238.

[50] D. Whitley, D. Hains and A. E. Howe, A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover, Parallel Problem Solving from Nature, PPSN XI, 6238 (2010), 566-575.

[51] D. Whitley, D. Hains and A. Howe, Tunneling between optima: partition crossover for the traveling salesman problem, Proceedings of the 11thACM Annual Conference on Genetic and Evolutionary Computation, (2009), 915-922.

[52] Z. Xiao and J. Chen, Research on Path Optimization Based on Improved Adaptive Genetic Algorithm, 7th Int. Conf. Intell. Human-Machine Syst. Cybern. (2015), 207-209.

[53] L. Yin, X. Li, L. Gao and C. Lu, A New Improved Fruit Fly Optimization Algorithm for Traveling Salesman Problem (2016), 21-28.

[54] O. Yugay, I. Kim, B. Kim and F.I.S. Ko, Hybrid genetic algorithm for solving Traveling Salesman Problem with sorted population, Third IEEE International Conference on Convergence and Hybrid Information Technology, (2008).

[55] http://comopt.ifi.uniidelberg.de/software/TSPLIB95/.

[56] Xinchao Zhao and Xiao-Shan Gao, Affinity genetic algorithm, Journal of Heuristics 13(2) (2007), 133-150.