Self-organisation migration technique for enhancing the permutation coded genetic algorithm

K. Dinesh*

Department of Computer Science and Technology,
Madanapalle Institute of Technology and Science,
Madanapalle,Chittoor District, Andhra Pradesh, India
Email: dineshhumar@gmail.com
*Corresponding author

R. Rajakumar

Department of Computer Science and Engineering,
Madanapalle Institute of Technology and Science,
Madanapalle, Chittoor District, Andhra Pradesh, India
Email: ramukshare@gmail.com

R. Subramanian

Department of Computer Science and Engineering,
Pondicherry University,
Puducherry, India
Email: rsmanian.csc@pondiuni.edu.in

Abstract: Genetic algorithm (GA) is well-known optimisation algorithm for solving various kinds of the optimisation problems. GA is based on the evolutionary principles and effectively solves the large-scale problem. In addition, it incorporates the variety of hybrid techniques to achieve the best performance in complex problems. However, self-organisation is one of the popular model, which acquire global order from the local interaction among the individuals. The combined version of self-organisation and genetic algorithm are adopted to improve the performance in attaining the convergence. This paper proposes a bi-directional self-organisation migration technique for improving the genetic algorithm which achieves the convergence and well-balanced diversity in the population. The experimentation is conducted on the standard test-bed of travelling salesman problem and instances are obtained from TSPLIB. Thus, the proposed algorithm has shown its dominance with the existing classical GA in terms of various parameter metrics.

Keywords: genetic algorithm; self-organisation migration algorithm; hybrid genetic algorithm; travelling salesman problem; TSP; pattern replacement; combinatorial problem.

Reference to this paper should be made as follows: Dinesh, K., Rajakumar, R. and Subramanian, R. (2021) 'Self-organisation migration technique for enhancing the permutation coded genetic algorithm', *Int. J. Applied Management Science*, Vol. 13, No. 1, pp.15–36.

Biographical notes: K. Dinesh is working as an Assistant Professor at the Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India. He completed his PhD at Pondicherry University and his areas of research are optimisation using nature-inspired algorithm, evolutionary algorithms, and cloud computing.

R. Rajakumar is working as an Assistant Professor at the Department of Computer Science and Engineering, Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, India. His research areas include evolutionary algorithms, wireless sensor network and soft-computing.

R. Subramanian is a Senior Professor in the Department of Computer Science, Pondicherry Central University, India. His specialisation includes robotics and evolutionary algorithms.

This paper is a revised and expanded version of a paper entitled 'Bi-directional self-organization technique for enhancing the genetic algorithm' presented at 'ICIT 2018 Proceedings of the 6th International Conference on Information Technology: IoT and Smart City', Hong Kong, 29–31 December 2018.

1 Introduction

Genetic algorithm (GA) is one of the popular optimisation algorithms which is inspired from the nature evaluation (Abramson et al., 1994; Ali et al., 2018; Fogarty and Huang, 1990; Latchoumi et al., 2019). The classical GA consists of several phases and among those, initialisation 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 initialisation is performed by random initialisation methods, which may also produce larger quantity of worst individuals (Yugay et al., 2008; Toğan and Daloğlu, 2008; Whitley et al., 2010; Sombat et al.., 2018). On the other hand, induced models of population seeding techniques lead the belated achievements of the fitness functions with poor convergence values (Schmitt and Amini, 1998; Kang and Jung, 2006; Paul et al., 2013b; Zhou et al., 2018). This tradeoff motivated the researchers to find new sets of population initialisation techniques for improving the phase specific as well as overall performance of the family of the GAs. In this perception, new mixture of initialisation methods that combine the benefits of both random and induced initialisation have been proposed by many of the researchers (Paul et al. (2013b, 2014; Chen and Liu, 2018).

It is widely demonstrated that 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. GA aims to find the optimum solution which is near to the objective function of the optimisation problems with very large search space (Katayama et al., 2000; Paul et al., 2014; Dinesh et al., 2017; Manicassamy et al., 2018).

2 Related work

Natural computing models and algorithms are frequently popular because of relative ease of implementation, wide applicability and reasonable quality of results delivered for many optimisation problems. Inspired by nature, researchers see the possibility of creating computational models based on biological processes and phenomena. Natural computing thus, involves the extraction of ideas from nature to the design of nature-inspired algorithms for solving complex problems (De Castro, 2007). In the recent years, the researchers proposed various kinds of novel computational methods for solving complex real-world problems. Many problems remained unsolved or poorly solved, opening the field for research of innovative solutions. Bio-inspired computing is one of the three categories of the broad field of natural computing (De Castro, 2006),which emerged from the idea of using nature's inspiration to search for new computational solutions to complex problems. It is fundamentally based on the behaviours of the living organisms, such as foraging, survival and self-organisation. These algorithms demonstrated their efficiency in various platforms including, complex problem solving and decision-making domains. Due to their importance and quality characteristics, the development of various biologically inspired computational algorithms has been recognised as one of the important advances made in the field of science and engineering. This class of Bio-inspired algorithms could be called as swarm intelligence (SI) algorithms, which is primarily based on the group behaviour of the living organisms. Variety of groups of algorithms are proposed in the family of Bio inspired algorithms such as particle swarm optimisation (PSO) (Eberhart and Kennedy, 1995; Nouiri et al., 2018), cuckoo search (Yang and Deb, 2009; El Aziz and Hassanien, 2018; Du et al., 2018), bat algorithm (Yang, 2010b; Dao et al., 2018; Liang et al., 2018) firefly algorithm (Yang, 2010a; Wang et al., 2018; Castillo et al., 2018) and so on, and the set of algorithms inspired by ant behaviour is one among them, which have been initially proposed and popularly evolved as ant colony optimisation (ACO) (Dorigo et al., 2006; Gülcü et al., 2018; Agrawal and Kaur, 2018; Rey et al., 2018).

Self-organising is a dynamic process which is not imposed by the external influence. In self-organisation, individuals interact directly to produce the common pattern which influences the behaviour of low level individual in the population individuals (Yugay et al., 2008; Toğan and Daloğlu, 2008; Whitley et al., 2010). Self-organising, often defined as, global ordering emerging from local interactions (Kallel and Schoenauer, 1997). In random immigrant technique, individuals interact between themselves and find the individual near to optimal solution and placed in subpopulation. Self-organisation and GA are combined in order to avoid the premature convergence and get trapped from the local optimum. The self organisation models enable the systems to acquire and maintain the structure by themselves, without any external control. It is highly evidenced that it gives a 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 (Jeong and Lee, 1998; Aras et al., 1999; Deep et al., 2008).

The self-organisation process have different kinds of interaction between the individuals in order to form global order from the local interaction (Al-Mulhem and Al-Maghrabi, 1998; Angeniol et al., 1988; Van Hulle, 2012). It is one of the important behaviour of self-organisation process. The characteristics of self-organisation behaviour of the work without any control action from the outside sources. The importance of the self-organisation process that can be observed from the behaviour is to avoid the situation of getting trapped with the local optimal solution (Tinós and Yang, 2007; Ray et al., 2007). The self organisation system improves the performance of the GA. Thus the self organisation process increases the researcher to use self-organisation for the optimisation of the complex problem. It also improves the performance of the GA in solving the optimisation and combinatorial problem. Many types of SOM algorithms (Jeong and Lee, 1998; Angeniol et al., 1988; Leung et al., 2004; Deep et al., 2008) have been proposed to solve the travelling salesman problem (TSP) by incorporating the efficient initialisation method and parameter adoption. Leung et al. (2004) 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. Jeong and Lee (1998) proposed the multimodal function optimisation, 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 Tinós and Yang (2007). According to dominant individuals and cyclical mutation operator, mutation probability will be tuned periodically. Although it offers better optimisation result, it suffered by faster convergence rate, which lead to end up with premature convergence. As another development, self-organising algorithm (migration algorithm) and GA are combined in order to avoid the premature convergence and get trapped from the local optimum (Tinós and Yang, 2007; Deep et al., 2008). 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-organising (migration algorithm) and GA gives different parameter design and setting, such as population size, crossover probability and mutation probability.

In its series, a new replacement technique has been proposed in Tajani et al. (2017a) that it replaces the worst individual of the population by random immigrants. These random immigrants 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. The novel weighted superposition-based meta-heuristic optimisation algorithm has been proposed by Baykasoğlu and Akpinar (2015) and it is inspired by principle of superposition of particles. The individuals in the population cooperatively generate the superposition which is further attracted by the remaining individuals in the population. This enhancement has utilised to solve the unconstrained, constrained (Baykasoğlu and Akpinar, 2017) and binary optimisation problem (Baykasoğlu and Ozsoydan, 2018). Further, this weighted superposition attraction helps to solve the local optima and premature convergence problem. Further, Karunanidy et al. (2017) proposed the new optimisation algorithm based on the monkey behaviour for balancing the exploration and exploitation. Similarly, the SOM has been efficiently utilised for improving the performance of the artificial raindrop algorithm in terms of diversity among the population (Jiang et al., 2018; Gu and Cheung, 2018). In most of the literature (Somhom et al., 1999; Kita et al., 2010; Popovic et al., 2014; Tajani et al., 2017b; Jiang et al., 2018), it is observed that self-organisation models are proposed to enhance the diversity among the population. So, in order to attain the convergence and also diversity in the population, we propose the novel bi-directional self organisation model to be incorporated with the classical GA, in order balance the search process between the convergence and the diversity among the population.

3 Model formation

The individuals in the population are primarily classified into two types namely, individuals with dominant fitness qualities are called as Leader individuals (θ^L) and remaining individuals are termed as active individuals (θ^A). Then the common patterns are generated by the interaction of individuals from the set leader individuals and similarly, the active individual generated the common sequence from selected p individuals. After this, the active individuals in the population are incorporating the intended sequence (common pattern) found from the leader individuals. Bidirectional components transfer, i.e., exchange of components, is being used here in order to improve the diversity among the individual from the population, which means, the sequence from the active individuals from the top ranked individuals from the set θ^A are incorporated towards the leader individuals (θ^L) from the population.

In the bi-directional self-organisation migration strategy, the bidirectional components have been utilised for enhancing the GA in terms of convergence and diversity among the individual from the population. Here, the TSP as selected test bed, in which the ultimate goal is that the salesman has to travel and visit every city in the tour and return to the starting city with minimal cost, where the cost may be the total distance of the tour. The TSP could be defined as follows: Let, the graph $G = (C, E)$ is an undirected graph with n cites $\{c_1, c_2, \ldots, c_n\}$ and the search space G is represented as: $\mathbb{G} = \frac{(n-1)!}{2}$.

Let the individuals θ are represented with the permutation of n cities as follows:

$$\theta\{c_1, c_2, \ldots, c_n\} \tag{1}$$

Let $POP_{unsorted}$ be the raw set of individuals for a population set and it can be defined as,

$$POP_{unsorted} = \{\theta_1, \theta_2, \ldots, \theta_{popsize}\} \tag{2}$$

where, *popsize* indicates the total number of individuals in the population.

Then, the fitness evaluation of individual in the population, $\forall\theta \in POP$ w.r.t. the TSP is defined as,

$$F(\theta) = \sum_{j=1}^{n-1} Dist(c_i, c_{i+1}) + Dist(c_n, c_1) \tag{3}$$

where

n	refers to the total number of cities
$Dist(c_i, c_{i+1})$	indicates the distance between city i and city i+1
$Dist(c_n, c_1)$	indicates to the distance between last city and first city to form complete tour.

With this background, the design may be continued by considering the equation (3) as the reference. After sorting, the population set would be defined as,

$$POP_{sorted} = \{\theta_1, \theta_2, \ldots, \theta_{popsize}\} \tag{4}$$

such that $F(\theta_1) \leq \_ F(\theta_2) \leq \ldots \leq F(\theta_{popsize})$.

Definition 1: The population are subdivided into leader and active individuals based on the fitness evaluation. The leader θ^L and active θ^A individuals are described as,

$$\theta^L = \{\theta_1, \theta_2, \ldots, \theta_p\} \tag{5}$$

$$\theta^A = \{\theta_{p+1}, \theta_{p+2}, \ldots, \theta_{popsize}\} \tag{6}$$

$$\theta^{A_temp} = \{\theta_{p+1}, \theta_{p+2}, \ldots, \theta_q\} \tag{7}$$

such that $\theta^{A_{temp}} \subset \theta^A,\ \theta^L \cup \theta^A = \varnothing : |\theta^L| \leq |\theta^A|, |\theta^L| = p,\ \theta^{A_{temp}} = q.$

Definition 2: The common pattern are generated from the set θ^L and $\theta^{A_{temp}}$. The sequence is defined as the set of similar element with identical order in two different individuals. The pattern obtained by performing tradeoff between two distinct individuals from the set θ^L or $\theta^{A_{temp}}$.

$$Pat_{i,j} = \theta_i \frac{order\ of\ visit}{\cap} \theta_j \tag{8}$$

such that $\forall_i, j \in \theta^L$ or $\theta^{A_{temp}}$ where, the set $Pat = \{c_1, c_2, \ldots, c_l\}$ be continuous common pattern attained from the above process. Then the length of the sequence is $|\ Pat\ | = l$ where, $l \leq n$.

The common pattern *Pat* obtained from equation (8) is found to be largest continuous pattern from the individuals. The empirical study pose that sequence utilised for replacement process may lead to the local optima. Therefore, the sequence can be further decomposes into smaller size with length '*r*' adopts to improve the results. Thus, the decomposed sequence maintained with length r and the equation (8) can be reconstructed as,

$$\begin{aligned} Pat_{s,r} &= Decompsed\left\{\frac{Pat(i,\ j)}{|Pat(i,\ j)|}\right\} \\ &= \{(c_i, c_{i+1}, c_{i+3}, c_r), (c_{r+1}, c_{r+3}, c_{r+4}, c_l)\} \end{aligned} \tag{9}$$

where, each *Pat* is decomposed into s number of sub-patterns.

Definition 3: The sub-pattern obtained from the equation (9) can be incorporated to the individuals, in the viable direction. The incorporated sub-pattern usually influence individual in the direction of dominant individuals, which proportionally amplify the individual. Consider, an sub-pattern $Pat_{s,r}$ has to integrate with the individual θ_k, which generally satisfy the sub-pattern of size 'r–1' as $\{c_i, c_{i+1}, c_{i+3}, c_{r-1}\}$ in θ_k.

$$\begin{Bmatrix} Incorporate\left(Pat_{s,r}, \theta_k\right), & \text{if } Pat_{s,r-1} \xrightarrow[\cap]{order\ of\ visit} \theta_k \\ Discard, & \text{otherwise.} \end{Bmatrix} \tag{10}$$

where the individual θ_k incorporates the sub-pattern $Pat_{s,r}$ using *Incorporate*($Pat_{s,r}$, θ_k) if it has the matching sub-pattern of size 'r – 1'.

In the incorporate process, the last element c_r from the sub-pattern is identified from the θ_k and it should be adjoined with the individual using the heuristic to form the sub-pattern. For example, consider the subpattern $Pat_{s,r}$ = {5, 4, 3, 2} and the individual θ_k = (9, 7, 1, 5, 4, 3, 6, 10, 2, 8) which as the identical sub-pattern of 5,4,3. The heuristic is as follows:

Case 1 The last element in the sub-pattern *cr* as 2 and it is adjoined with the available sub-pattern, such that the pattern of {5, 4, 3, 2} is present in θ_k also in the same order and it will become as θ_k = (9, 7, 1, 5, 4, 3, 2, 6, 10, 8).

Case 2 The position of 5, 4, 3 must be changed such that it should be adjoined with the last element 2, such that the sequence of 5, 4, 3, 2 is present in θ_k also in the same order and it will become as θ_k = (9, 7, 1, 6, 10, 5, 4, 3, 2, 8).

Definition 4: The set of dominant individuals with best fitness value is known as elitist individual (θ^E). It could be described as,

$$\theta^E = \{\theta_1, \theta_2, \ldots, \theta_c\} \tag{11}$$

such that $F(\theta_1) \leq F(\theta_2) \leq \ldots \leq F(\theta_e)$, $|\theta^E| = e$.

4 Proposed algorithm

Based on the above formulation procedure, the bi-directional self-organised genetic algorithm (BDSOGA) algorithm is being devised as explained in Algorithm 1. The initial population could be generated using any one of the three population seeding techniques namely random, nearest neighbour (NN) and Vari-begin with variable diversity. The individuals are evaluated and sorted according to the fitness value and the similar patterns present in the leader individuals are found. Then active individuals are moved towards the leader individuals by incorporating the similar pattern of leader individuals.

Algorithm 1 Proposed bi-directional self-organised genetic algorithm

Assumption

Let *p* and *q* it is fixed as 10. The *r* value is fixed as 4 for cites $1 \leq n \leq 500$ otherwise *r* is 6 and *e* as 4. The crossover probability Pc value and mutation probability Pm is fixed a 0.7 and 0.1.

Step 1: Initialisation.

a Choose the appropriate TSP instance from TSPLIB.

b Generate the initial population set using anyone of population initialisation technique.

Step 2: Evaluate the fitness of each individual using the equation (3).

Step 3: Get the sorted population set POP_{sorted} from equation (4).

Step 4: Derive the sets θ^L, θ^A and $\theta^{A_{temp}}$ as described in the equations (5), (6) and (7).

Step 5: Find the common sequence in θ^L.

a Choose two different individuals as θ_g^L and θ_h^L from the set θ^L.

b Select c_i from θ_g^L and retrieve the positions of c_i in θ_h^L.

x = position of c_i in θ_g^L

y = position of c_i in θ_h^L

c While ($C[x] == C[y]$) // Continue upto the similar elements

{$x = x + 1$ and $y = y + 1$} // Increment the position values until the dissimilar elements found

$x = x - 1$ and $y = y - 1$ // Fixing the pointer values at the last matching element

d Derive the sets $Pat_{s,r}^L$ as described in the equations (8) and (9).

e Repeat through the Step 5(a) until $g \le p$ else goto Step 5(f).

f Repeat through the Step 5(a) until $h \le p$ else return $Pat_{s,r}^L$.

Step 6: Migrate the θ^A towards θ^L from equation (10).

Step 7: Find the common sequence in $\theta^{A_{temp}}$.

a Choose two different individuals as $\theta_g^{A_{temp}}$ and $\theta_h^{A_{temp}}$ from the set θ^L.

b Select c_i from $\theta_g^{A_{temp}}$ and retrieve the positions of c_i in $\theta_h^{A_{temp}}$.

x = Position of c_i in $\theta_g^{A_{temp}}$

y = Position of c_i in $\theta_h^{A_{temp}}$

c While ($C[x] = = C[y]$) // Continue upto the similar elements in both the individuals

{$x = x + 1$ and $y = y + 1$} // Increment the position values until the dissimilar elements found

$x = x - 1$ and $y = y - 1$ // Fixing the pointer values at the last matching element

d Derive the sets $Pat_{s,r}^A$ as described in the equations (8) and (9).

e Repeat through the Step 7(a) until $g \le p$ else goto Step 7(f).

f Repeat through the Step 7(a) until $h \le p$else return $Pat_{s,r}^A$.

Step 8: Migrate the θ^L towards $\theta^{A_{temp}}$ from equation (10).

Step 9: Rebuild the population by combining $\theta^L \cup \theta^A$.

Step 10: Evaluate the fitness of each individual in the POP using equation (3) and sort population with respect to fitness value using equation (4).

Step 11: Export the elitist Individuals according to the equation (11) to the next generation.

Step 12: Perform greedy crossover with probability Pc.

Step 13: Perform swap mutation with probability Pm.

Step 14: Repeat through Step (2) until the termination condition.

Then similar pattern from the active individuals are found and the pattern is incorporated in the leader individuals of the population. The superior individual is selected for next generation according to elitist rate y. Crossover operation is performed to all the individuals in the population with crossover probability (Pc) to produce the new individuals. Mutation operator with mutation probability (Pm) is finally applied.

The population initialisation technique plays an important role in providing the good quality individual for the population which improves the optimal solution. The individuals (θ) are represented as (c_1, c_2, c_3, …, c_n) cities, which is a permutation of (1, 2, …, n). In the random technique, the individual is formed by choosing the first city c_1 in a random manner. In order to select the next city c_j where $c_j \notin \theta_i$ then this process continues until individual is created. In the NN technique, the initial population is created 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_n where $cj \notin \theta_i$ then visit c_j. The initial population POP is created using the same process. Initial population is created by using the Vari-begin with variable diversity (VV) by order distance vector (ODV).

In VV (ODV) population initialisation 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 $c_j \notin \theta_i$ then visit c_j. This process continues until all the individuals for the population (POP) are generated. Then select the leader individual $\{\theta_1, \theta_2, \ldots, \theta_p\}$ from the population where leader individuals of set θ^L are having the highest fitness values in population and the remaining individuals $\{\theta_{p+1}, \theta_{p+2}, \ldots, \theta_{popsize}\}$ from the population are active individuals of set θ^A. Then, the set $\theta^{A_{temp}}$ are also derived from the set $\theta^{A_{temp}} = \{\theta_{p+1}, \theta_{p+2}, \ldots, \theta_q\}$.

Patterns are discovered from the leader individuals (θ^L) by comparing the two leader individual θ_g^L and θ_h^L from the set θ^L. The pattern $Pat_{s,r}^L$ is the set of cities visited in same order by the individual θ_g^L and θ_h^L from the set θ^L. The patterns are of length s. These patterns are interpreted as the sub-tour of the individuals. The individuals are formed with sub-tour having high fitness when compared to other individuals. In order to find the patterns which commonly match between two leader individual, the first city (c_1) is selected from the leader individual θ_g^L and marked its position as x in θ_h^L and then find the city (c_1) in θ_h^L and mark its position as y. Then increment the positions of x and y by 1 and check the city on positions x and y whether there is a match between both the individual (θ_g^L and θ_h^L). If the city matches then repeat the same step until the sequence ($Pat_{s,r}^L$) of length s is formed. Otherwise, increment the next city in θ_g^L and continue with same steps.

The similar pattern $Pat_{s,r}^A$ from the active individuals $\theta^{A_{temp}}$ are used to improve the diversity of the individuals from the set θ^L. In order to find the common sequence from the active individuals we have followed the same procedure to find the sequence from the leader individuals. Then the pattern is used to increase the diversity among the individuals present from the set $\theta^{A_{temp}}$. Then the greedy crossover technique, which sometimes referred as greedyswap is used here. 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 Experiments and discussion

The performance of the proposed algorithm is manifested using a standard test-bed. Travelling salesman problem (TSP), a subset of NP-hard problem which falls under the category of combinatorial optimisation is selected as the test-bed (Whitley et al., 2009;

Okano et al., 1999) and obtained from TSPLIB (Reinelt, 2012). The set of instances such as eil51, pr76, kroA100, pr144, fl417 and u724 are selected for experimentation process.

The experiments are conducted in three different phases: *Phase* 1 is used to analysis the performance of the proposed bi-directional self-organised random (BDSOR) in conjunction with the random (R) population initialisation technique (Ahn et al., 2003; Balling et al., 2006; Kallel and Schoenauer, 1997) in a classical GA; *Phase* 2 is used to assess the performance of the proposed bi-directional self-organised NN (BDSONN) in colligation with the NN population seeding technique Lu et al. (2004); Ray et al., 2007) in a classical GA; *Phase* 3 is used to measure the superiority of the proposed bi-directional self-organised VV (BDSOVV) in conjugation with the VV population initialisation technique (Paul et al., 2014, 2013a) in a classical GA.

The size of population is fixed as 100 and the number of generations are set as 100 for all the instances. For each case, 50 independent runs is considered with the crossover probability as 0.7 ($P_c = 0.7$) and the mutation probability as 0.1 ($P_m = 0.1$).

5.1 Evaluation metric

The different types of evaluation metrics for investigating the performance of proposed algorithm with the GA. The various types of performance metric used for analysing the potential of proposed algorithm are convergence, average convergence, error rate, distinct individual, convergence diversity and NN ratio. The metrics are selected to analysis the performance of the proposed algorithm in terms of convergence towards the optimal solution, diversity maintained among the individual in the population and also the overall convergence of the population.

5.2 Result analysis

The experimental set-up in terms of TSP instances, performance assessment factors, the simulation runs is as same as designed above. The results of the experiments are presented in the Tables 1 to 3 and the corresponding analyses are illustrated in the Figures 1 to 7.

5.2.1 Phase-1: random (R) vs. bi-directional self-organised random

Technique In this first experimental phase, we have used the random initialisation techniques for generating the initial population for the classical GA and BDSOGA. In the random seeding techniques, the individuals are formed in the random techniques (i.e.) the cities are visited in the random manner without following any heuristics rules. The performance of both the existing and proposed techniques is analyzed using the standard GA parameter as mention above. Table 1 clearly shows the performance of both the classical random and BDSOR. From the table observation, the convergence rate of self-organisation technique dominates the classical GA. The convergence rate of BDSOR technique has the highest convergence of 84.43% for the instance pr76 and the lowest convergence value of 72.21%for the instance kroA100. Similarly the best fitness of the instance rat99 is 1,480.14 and 1,434.82 for the random and BDSOR technique. Likewise for the worst and average fitness value of the random technique is strongly dominated by the proposed BDSOR technique.

Table 1 Random vs. BDSOR technique

S. no	*Instance*	*Technique*	*Time*	*Opt. val*	*Fitness*			*Conv. rate (%)*		*Error rate (%)*		*Avg. C. (%)*	*NN(%)*	*D. ind.*	*Conv. div.*	
					Best	*Worst*	*Average*	*Best*	*Worst*	*Best*	*Worst*				*Best*	*Average*
1	eil51	R	43.49	426	522.44	1,022.53	890.66	77.36	–40.03	22.64	140.03	–9.08	16.08	85	117.39	86.44
		BDSOR	361.73	426	502.13	920.7	800.35	82.13	–16.13	17.87	116.13	12.12	30.57	82	98.26	70.01
2	pr76	R	70.39	108,159	131,082.8	239,676.2	218,945.99	78.81	–21.6	21.19	121.6	–2.43	7.64	82	100.4	81.24
		BDSOR	653.02	108,159	125,000.88	217,592.66	200,475.99	84.43	–1.18	15.57	101.18	14.65	26.96	81	85.61	69.78
3	rat99	R	92.42	1,211	1,480.14	2,897.43	2,590.06	77.78	–39.26	22.22	139.26	–13.88	5.65	88	117.03	91.65
		BDSOR	504.54	1,211	1,434.82	2,628.18	2,347	81.52	–17.03	18.48	117.03	6.19	24.32	84	98.54	75.32
4	kroA100	R	93.56	21,282	27,883.73	52,237	44,651.89	68.98	–45.45	31.02	145.45	–9.81	3.72	78	114.43	78.79
		BDSOR	1,830.71	21,282	27,196.96	45,801.49	41,407.63	72.21	–15.21	27.79	115.21	5.43	31.09	77	87.42	66.77
5	pr144	R	141.12	58,537	71,188.74	149,016.52	125,840.3	78.39	–54.57	21.61	154.57	–14.98	2.22	85	132.95	93.36
		BDSOR	1,079.15	58,537	70,923.13	130,028.74	113,902.46	78.84	–22.13	21.16	122.13	5.42	23.35	84	100.97	73.42
6	tsp225	R	229.07	3,919	4,803.16	10,361.91	8,953	77.44	–64.4	22.56	164.4	–28.45	1.05	81	141.84	105.89
		BDSOR	4,116.21	3,919	4,610.43	9,361.85	7,616.89	82.36	–38.88	17.64	138.88	5.64	21.18	80	121.24	76.72
7	fl417	R	483.45	11,861	14,945.57	34,181.6	27,387.99	73.99	–88.18	26.01	188.18	–30.91	0.81	92	162.18	104.9
		BDSOR	6,125.68	11,861	14,331.58	29,242.13	24,277.82	79.17	–46.54	20.83	146.54	–4.69	16.73	90	125.71	83.86
8	u724	R	944.55	41,910	54,068.09	121,805.62	99,013.84	70.99	–90.64	29.01	190.64	–36.25	0.31	87	161.63	107.24
		BDSOR	11,392.4	41,910	52,994.33	109,460.89	87,593.45	73.55	–61.18	26.45	161.18	–9	14.11	83	134.73	82.56

Notes: D. ind. – distinct individual, con. div. – convergence diversity, NN – nearest neighbour rate, avg. C. – average convergence rate, opt. value – optimum value.

Table 2 NN vs. bi-directional self-organised NN technique

S. no	*Instance*	*Technique*	*Time*	*Opt. val*	*Fitness*			*Conv. rate (%)*		*Error rate (%)*		*Avg. C. (%)*	*NN(%)*	*D. ind.*	*Conv. div.*	
					Best	*Worst*	*Average*	*Best*	*Worst*	*Best*	*Worst*				*Best*	*Average*
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	46.17	32.24
		BDSONN	270.47	426	470.57	764.13	606.91	89.54	20.63	10.46	79.37	57.53	53.22	79	68.91	32.01
2	pr76	NN	61.01	108,159	122,000.59	170,440.39	165,084.5	87.2	42.42	12.8	57.58	47.37	59.45	45	44.79	39.83
		BDSONN	526.44	108,159	121,226.44	179,309.3	149,087.37	87.92	34.22	12.08	65.78	62.16	61.9	73	53.7	25.76
3	rat99	NN	67.03	1,211	1,351.05	2,065.92	1,910.6	88.44	29.4	11.56	70.6	42.23	62.82	34	59.03	46.21
		BDSONN	430.68	1,211	1,294.82	2,144.38	1,895.21	93.08	22.92	6.92	77.08	43.5	50.59	71	70.15	49.58
4	kroA100	NN	70.4	21,282	23,727.76	36,985.82	34,333.56	88.51	26.21	11.49	73.79	38.67	62.68	37	62.3	49.83
		BDSONN	1,726.3	21,282	23,510.84	37,528.41	34,047.75	89.53	23.66	10.47	76.34	40.02	52.9	74	65.87	49.51
5	pr144	NN	94.08	58,537	64,808.37	104,615.59	90,970.95	89.29	21.28	10.71	78.72	44.59	76.04	34	68	44.69
		BDSONN	1,192.3	58,537	62,513.34	108,545.21	98,173.69	93.21	14.57	6.79	85.43	32.29	67.1	69	78.64	60.92
6	tsp225	NN	160.65	3,919	4,619.35	6,864.59	6,296.46	82.13	24.84	17.87	75.16	39.33	68.09	39	57.29	42.79
		BDSONN	3,303.9	3,919	4,556.86	7,358.89	5,920.54	83.72	12.23	16.28	87.77	48.93	54.54	77	71.5	34.8
7	fl417	NN	339.84	11,861	14,597	21,721.28	19,385.66	76.93	16.87	23.07	83.13	36.56	63.75	34	60.06	40.37
		BDSONN	6,428.3	11,861	13,154.25	21,664.39	18,352.59	89.1	17.35	10.9	82.65	45.27	50.61	75	71.75	43.83
8	u724	NN	593.16	41,910	51,210.9	72,857.82	70,221.87	77.81	26.16	22.19	73.84	32.45	66.66	33	51.65	45.36
		BDSONN	10,329	41,910	47,452.7	75,339.87	64,374.27	86.77	20.23	13.23	79.77	46.4	54.032	82	66.54	40.38

Notes: D. ind. – distinct individual, con. div. – convergence diversity, NN – nearest neighbour rate, avg. C. – average convergence rate, opt. value – optimum value.

Table 3 VV vs. bi-directional self-organised VV technique

S. no	*Instance*	*Technique*	*Time*	*Opt. val*	*Fitness*			*Conv. rate (%)*		*Error rate (%)*		*Avg. C. (%)*	*NN(%)*	*D. ind.*	*Conv. div.*	
					Best	*Worst*	*Average*	*Best*	*Worst*	*Best*	*Worst*				*Best*	*Average*
1	eil51	VV	46.27	426	453.7	585.78	507.44	93.5	62.49	6.5	37.51	80.88	47.33	74	31	12.61
		BDSOVV	305.47	426	446.87	627.25	493.44	95.1	52.76	4.9	47.24	84.17	59.68	58	42.34	10.93
2	pr76	VV	65.21	108,159	113,657.75	160,440.39	128,063.45	94.92	51.66	5.08	48.34	81.6	53.67	48	43.25	13.32
		BDSOVV	545.93	108,159	112,217.09	178,852.6	130,887.62	96.25	34.64	3.75	65.36	78.99	57.14	58	61.61	17.26
3	rat99	VV	89.37	1,211	1,317.56	1,765.92	1,523.41	91.2	54.18	8.8	45.82	74.2	52.6	51	37.02	17
		BDSOVV	448.43	1,211	1,291.18	1,988.34	1,537.35	93.38	35.81	6.62	64.19	73.05	56.9286	59	57.57	20.33
4	kroA100	VV	90.56	21,282	22,539.64	29,985.82	25,973.4	94.09	59.1	5.91	40.9	77.96	52.2	50	34.99	16.13
		BDSOVV	1,847.5	21,282	22,490.86	37,747.56	27,192.77	94.32	22.63	5.68	77.37	72.23	54.31	58	71.69	22.09
5	pr144	VV	123.56	58,537	61,663.21	84,615.59	72,425.48	94.66	55.45	5.34	44.55	76.27	62.01	55	39.21	18.39
		BDSOVV	937.06	58,537	61,609.16	103,468.97	73,041.55	94.75	23.24	5.25	76.76	75.22	56.3217	66	71.51	19.53
6	tsp225	VV	205.45	3,919	4,610.68	5,864.59	4,905.71	82.35	50.36	17.65	49.64	74.82	51.71	60	32	7.53
		BDSOVV	3,441.9	3,919	4,294.48	6,750.08	5,066.25	90.42	27.76	9.58	72.24	70.73	52.67	54	62.66	19.69
7	fl417	VV	411.89	11,861	13,181.5	17,721.28	15,378.39	88.87	50.59	11.13	49.41	70.34	45.61	61	38.27	18.52
		BDSOVV	6,418.5	11,861	12,751.93	20,340.84	15,727.89	92.49	28.51	7.51	71.49	67.4	42.59	62	63.98	25.09
8	u724	VV	748.23	41,910	44,907.52	60,857.82	49,223.53	92.85	54.79	7.15	45.21	82.55	50.78	60	38.06	10.3
		BDSOVV	10,396	41,910	44,995.92	71,847.82	50,204.02	92.64	28.57	7.36	71.43	80.21	53.5228	64	64.07	12.43

Notes: D. ind. – distinct individual, con. div. – convergence diversity, NN – nearest neighbour rate, avg. C. – average convergence rate, opt. value – optimum value.

Figure 1 Best convergence rate (see online version for colours)

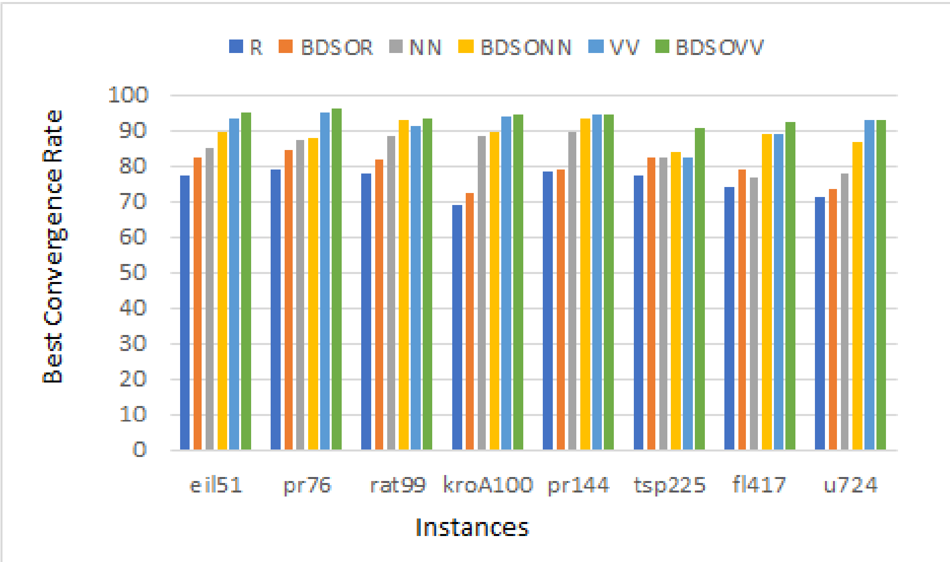


Figure 2 Worst convergence rate (see online version for colours)

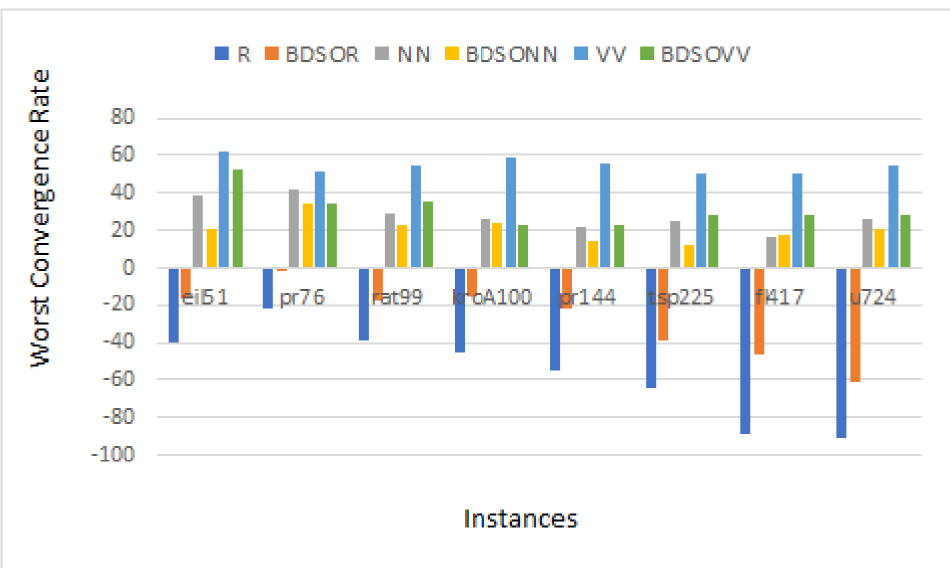


Figure 3 Average convergence rate (see online version for colours)

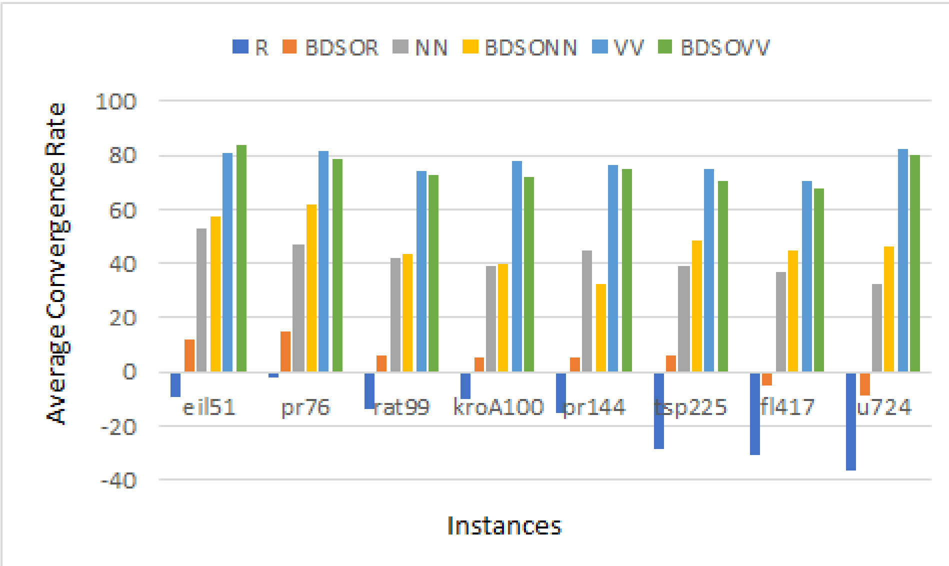


Figure 4 Best convergence diversity (see online version for colours)

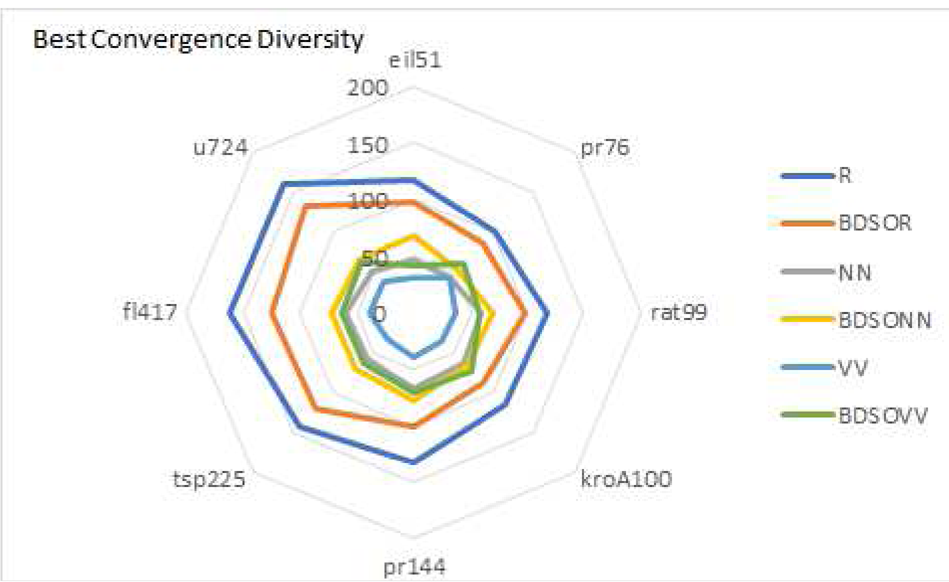


Figure 5 Average convergence diversity (see online version for colours)

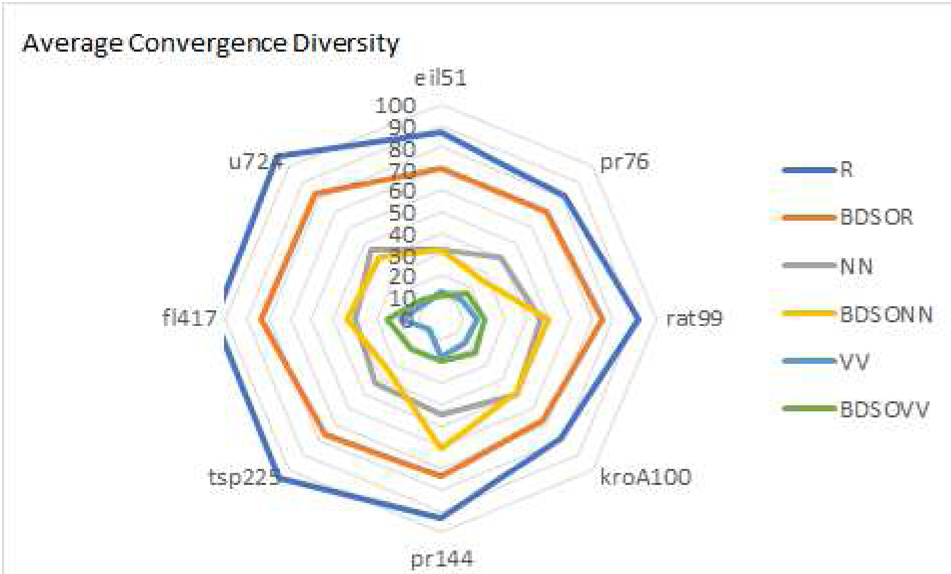


Figure 6 NN ratio (see online version for colours)

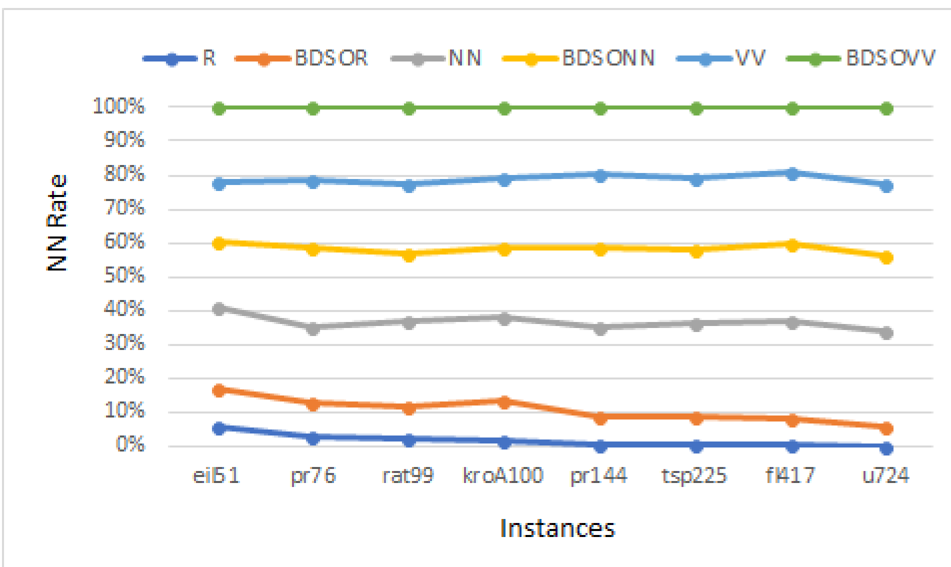
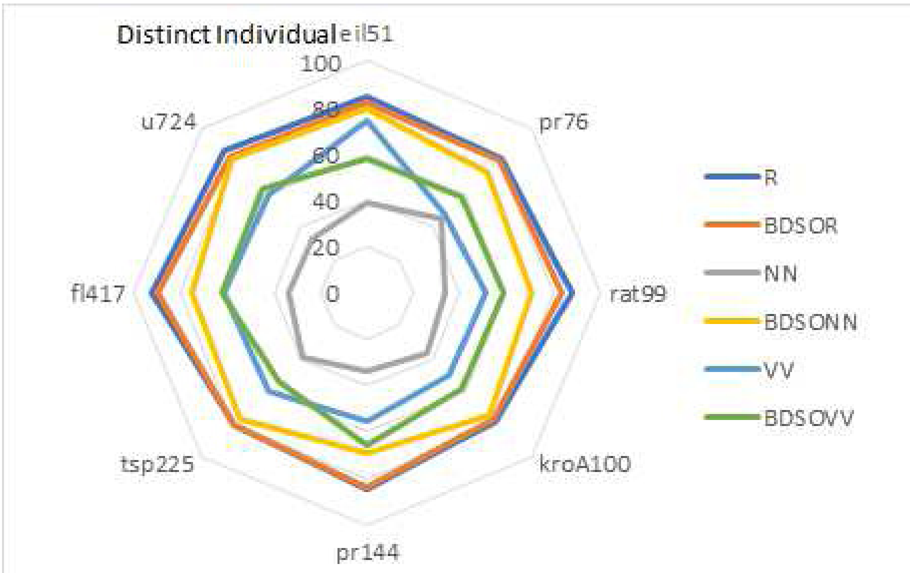


Figure 7 Distinct individual (see online version for colours)



Hence by using the same instance, the worst convergence rate of the random technique is –39% and –17% for the BDSOR technique. The proposed technique achieves the best and worst error rate of 18.48 and 117.03 respectively, but the random technique has the value of 22.22 and 139.26 for the same. Congruently in terms of distinct individual, the random technique has 88 distinct individual in the population whereas the BDSOR has the less number of distinct individual as 84. The best and average convergence diversity of the BDSOR is 98.54 and 75.32 which is dominated by the random techniques that has the value 117.03 and 91.65 due to the random population seeding 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.

5.2.2 Phase 2: NN vs. BDSONN technique

NN technique and its self-organised rendition BDSONN are utilised for introductory population generation as a part of a classical GA in this stage. Here the choice of cities to be gone to is made as capacity of separation between the same. Inclination is given to the cities with least separation values and it is made by a situated of heuristics. The BDSONN technique generated heading execution regarding fitness value over the established NN technique. For instance, for the instance rat99, the average fitness value accomplished by the BDSONN technique as 1,895.21,while the other arrived at the average fitness value as 1,910.60. This points the superiority of the proposed technique against the existing one. Similarly, the investigating capacity of proposed technique is additionally demonstrated as far as the number of distinct individuals: the BDSONN technique processed the average of 71 distinct individuals, in as much as the existing one investigates just 34 distinct individuals, which clearly demonstrated the exploration capacity of the proposed model. Notwithstanding these, it is additionally watched that the proposed technique slacks regarding NN ratio (whose value is 62.82%) w.r.t. the BDSONN model (whose value is 50.59%). The best and worst convergence diversity of the proposed technique is 70.15 and 49.58 which has good performance when compared with the existing NN technique has the value 59.03 and 46.21. These investigates are undoubtedly demonstrated the capacity of the proposed technique in all perspectives.

5.2.3 Phase 3: VV vs. BDSOVV technique

This phase of experiments uses the VV technique and its variant BDSOVV as initialisation 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 1 and the corresponding results are presented in Table 3. As like in the previous experimental phases, the proposed BDSOVV technique presides over the existing VV technique. For the detailed discussion, by considering the sample instance eil51, it is observed that the former one is leading the later one by all convergence arte as 95.10%, whereas the existing VV technique accomplished the best convergence rate as 93.50% and the worst convergence rate is 62.49% respectively. Similarly, the BDSOVV technique produced the average convergence rate as 84.17%; conversely the VV technique produced the same as 80.88% 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 BDSOVV technique for the same instance eil51 is very less as the best error rate as 4.90%and the worst rate is around 47.24%; on the other hand the best error rate of the existing technique is about 6.50% and 37.51% respectively. It proves the effectiveness of the proposed technique in comparison with its existing version.

In the same way, the BDSOVV technique produced leading performance in terms of fitness value over the classical VV technique. For example, for the instance eil51, the average fitness value achieved by the BDSOVV technique as 493.44, while the other reached the average fitness value as 507.44. 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 BDSOVV technique produced the average of 74 distinct individuals, whereas the existing one explores only 58 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 47.33%) w.r.t. the BDSOVV model (whose value is 59.68%). These analyses are undoubtedly demonstrated the ability of the proposed technique in all aspects. The self-organised technique has the convergence diversity of 42.34 and the existing technique has the convergence diversity has 31.00. Likewise for the same the average convergence diversity of the VV technique has dominated the self-organised technique.

6 Conclusions and future work

The combination of self-organisation technique with GA has improved the performance with respect to convergence and convergence diversity. The experimentation process is conducted in three phases and all the phases have utilised a common set of experimentation setup. Experiments results illustrate the performance of the proposed algorithm with the competitor algorithm in all the three phases and obtained better results in term of various performance metrics. This shows that the proposed algorithm has a better exploration and exploitation to find the optimal solutions. Finally, the performance of the proposed algorithm clearly proved its dominance over the classical GA, which achieves the better convergence and also avoids the premature convergence by maintaining well-balanced diversity in the population. Thus the outcome of this work required further enhancement with respect to the computation time.

Acknowledgements

This work is a part of the Research Projects sponsored by Visvesvaraya PhD Scheme for Electronics and IT, Ministry of Electronics and Information Technology, India, and reference no. VISPHD-MEITY-2230, 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

Abramson, D., Mills, G. and Perkins, S. (1994) 'Parallelisation of a genetic algorithm for the computation of efficient train schedules', *Parallel Computing and Transputers*, Vol. 37, No. 1, pp.139–149.

Agrawal, A.P. and Kaur, A. (2018) 'A comprehensive comparison of ant colony and hybrid particle swarm optimization algorithms through test case selection', in *Data Engineering and Intelligent Computing*, pp.397–405, Springer, Singapore.

Ahn, C.W., Kim, K.P. and Ramakrishna, R.S. (2003) 'A memory-efficient elitist genetic algorithm', *International Conference on Parallel Processing and Applied Mathematics*. Springer, pp.552–559.

Ali, M.Z., Awad, N.H., Suganthan, P.N., Shatnawi, A.M. and Reynolds, R.G. (2018) 'An improved class of real-coded genetic algorithms for numerical optimization', *Neurocomputing*, Vol. 275, No. 1, pp.155–166.

Al-Mulhem, M. and Al-Maghrabi, T. (1998) 'Efficient convex-elastic net algorithm to solve the Euclidean traveling salesman problem', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 28, No. 4, pp.618–620.

Angeniol, B., Vaubois, G.D.L.C. and Le Texier, J-Y. (1988) 'Self-organizing feature maps and the travelling salesman problem', *Neural Networks*, Vol. 1, No. 4, pp.289–293.

Aras, N., Oommen, B.J. and Altınel, I. (1999) 'The kohonen network incorporating explicit statistics and its application to the travelling salesman problem', *Neural Networks*, Vol. 12, No. 9, pp.1273–1284.

Balling, R.J., Briggs, R.R. and Gillman, K. (2006) 'Multiple optimum size/shape/topology designs for skeletal structures using a genetic algorithm', *Journal of Structural Engineering*, Vol. 132, No. 7, pp.1158–1165.

Baykasoğlu, A. and Akpinar, S. (2015) 'Weighted superposition attraction (WSA): a swarm intelligence algorithm for optimization problems – part 2: constrained optimization', *Applied Soft Computing*, Vol. 37, No. 1, pp.396–415.

Baykasoğlu, A. and Akpinar, S. (2017) 'Weighted superposition attraction (WSA): a swarm intelligence algorithm for optimization problems – part 1: unconstrained optimization', *Applied Soft Computing*, Vol. 56, No. 1, pp.520–540.

Baykasoğlu, A. and Ozsoydan, F.B. (2018) 'Dynamic optimization in binary search spaces via weighted superposition attraction algorithm', *Expert Systems with Applications*, Vol. 96, No. 1, pp.157–174.

Castillo, O., Soto, C. and Valdez, F. (2018) 'A review of fuzzy and mathematic methods for dynamic parameter adaptation in the firefly algorithm', in *Advances in Data Analysis with Computational Intelligence Methods*, pp.311–321, Springer, Cham.

Chen, C-C. and Liu, Y-T. (2018) 'Enhanced ant colony optimization with dynamic mutation and ad hoc initialization for improving the design of tsk-type fuzzy system', *Computational Intelligence and Neuroscience*, Vol. 2018, pp.1–15.

Dao, T-K., Pan, T-S., Pan, J-S. et al. (2018) 'Parallel bat algorithm for optimizing makespan in job shop scheduling problems', *Journal of Intelligent Manufacturing*, Vol. 29, No. 2, pp.451–462.

De Castro, L.N. (2006) *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*, Chapman and Hall/CRC Press.

De Castro, L.N. (2007) 'Fundamentals of natural computing: an overview', *Physics of Life Reviews*, Vol. 4, No. 1, pp.1–36.

Deep, K. et al. (2008) 'A self-organizing migrating genetic algorithm for constrained optimization', *Applied Mathematics and Computation*, Vol. 198, No. 1, pp.237–250.

Dinesh, K., Amudhavel, J. and Subramanian, R. (2017) 'A novel self-organization model for improving the performance of permutation coded genetic algorithm', *Advances and Applications in Mathematical Sciences*, Vol. 17, No. 1, pp.1–29.

Dorigo, M., Birattari, M. and Stutzle, T. (2006) 'Ant colony optimization', *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp.28–39.

Du, X., Wang, J., Jegatheesan, V. and Shi, G. (2018) 'Parameter estimation of activated sludge process based on an improved cuckoo search algorithm', *Bioresource Technology*, Vol. 249, No. 1, pp.447–456.

Eberhart, R. and Kennedy, J. (1995) 'Particle swarm optimization', in *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, No. 1, pp.1942–1948.

El Aziz, M.A. and Hassanien, A.E. (2018) 'Modified cuckoo search algorithm with rough sets for feature selection', *Neural Computing and Applications*, Vol. 29, No. 4, pp.925–934.

Fogarty, T.C. and Huang, R. (1990) 'Implementing the genetic algorithm on transputer based parallel processing systems', *International Conference on Parallel Problem Solving from Nature*, Springer, pp.145–149.

Gu, F. and Cheung, Y-M. (2018) 'Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm', *IEEE Transactions on Evolutionary Computation*, Vol. 22, No. 2, pp.211–225.

Gülcü, S., Mahi, M., Baykan, Ö.K. and Kodaz, H. (2018) 'A parallel cooperative hybrid method based on ant colony optimization and 3-opt algorithm for solving traveling salesman problem', *Soft Computing*, Vol. 22, No. 5, pp.1669–1685.

Jeong, I-K. and Lee, J-J. (1998) 'A self-organizing genetic algorithm for multimodal function optimization', *Artificial Life and Robotics*, Vol. 2, No. 1, pp.48–52.

Jiang, Q., Wang, L., Hei, X., Cheng, J., Lu, X., Lin, Y. and Yu, G. (2018) 'Arae-som+ bco: an enhanced artificial raindrop algorithm using self-organizing map and binomial crossover operator', *Neurocomputing*, Vol. 275, No. 1, pp.2716–2739.

Kallel, L. and Schoenauer, M. (1997) 'Alternative random initialization in genetic algorithms', *ICGA*, pp.268–275.

Kang, R-G. and Jung, C-Y. (2006) 'The improved initializationmethod of genetic algorithm for solving the optimization problem', in *Neural Information Processing*, pp.789–796, Springer, Berlin, Heidelberg.

Karunanidy, D., Amudhavel, J., Datchinamurthy, T.S. and Ramalingam, S. (2017) 'A novel java macaque algorithm for travelling salesman problem', *IIOAB Journal*, Vol. 8, No. 2, pp.252–261.

Katayama, K., Sakamoto, H. and Narihisa, H. (2000) 'The efficiency of hybrid mutation genetic algorithm for the travelling salesman problem', *Mathematical and Computer Modelling*, Vol. 31, Nos. 10–12, pp.197–203.

Kita, E., Kan, S. and Fei, Z. (2010) 'Investigation of self-organizing map for genetic algorithm', *Advances in Engineering Software*, Vol. 41, No. 2, pp.148–153.

Latchoumi, T., Balamurugan, K., Dinesh, K. and Ezhilarasi, T. (2019) 'Particle swarm optimization approach for waterjet cavitation peening', *Measurement*, Vol. 141, No. 1, pp.184–189.

Leung, K-S., Jin, H-D. and Xu, Z-B. (2004) 'An expanding self-organizing neural network for the traveling salesman problem', *Neurocomputing*, Vol. 62, No. 1, pp.267–292.

Liang, H., Liu, Y., Shen, Y., Li, F. and Man, Y. (2018) 'A hybrid bat algorithm for economic dispatch with random wind power', *IEEE Transactions on Power Systems*, Vol. 33, No. 5, pp.5052–5061.

Lu, Y., Lu, S., Fotouhi, F., Deng, Y. and Brown, S.J. (2004) 'Incremental genetic k-means algorithm and its application in gene expression data analysis', *BMC Bioinformatics*, Vol. 5, No. 1, p.172.

Manicassamy, J., Karunanidhi, D., Pothula, S., Thirumal, V., Ponnurangam, D. and Ramalingam, S. (2018) 'GPS: a constraint-based gene position procurement in chromosome for solving large-scale multiobjective multiple knapsack problems', *Frontiers of Computer Science*, Vol. 12, No. 1, pp.101–121.

Nouiri, M., Bekrar, A., Jemai, A., Niar, S. and Ammari, A. C. (2018) 'An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem', *Journal of Intelligent Manufacturing*, Vol. 29, No. 3, pp.603–615.

Okano, H., Misono, S. and Iwano, K. (1999) 'New tsp construction heuristics and their relationships to the 2-opt', *Journal of Heuristics*, Vol. 5, No. 1, pp.71–88.

Paul, P.V., Dhavachelvan, P. and Baskaran, R. (2013a) 'A novel population initialization technique for genetic algorithm', *International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, IEEE, pp.1235–1238.

Paul, P.V., Ramalingam, A., Baskaran, R., Dhavachelvan, P., Vivekanandan, K., Subramanian, R., and Venkatachalapathy, V. (2013b) 'Performance analyses on population seeding techniques for genetic algorithms', *International Journal of Engineering and Technology (IJET)*, Vol. 5, No. 3, pp.2993–3000.

Paul, P.V., Ramalingam, A., Baskaran, R., Dhavachelvan, P., Vivekanandan, K. and Subramanian, R. (2014) 'A new population seeding technique for permutation-coded genetic algorithm: Service transfer approach', *Journal of Computational Science*, Vol. 5, No. 2, pp.277–297.

Popovic, D., Moschopoulos, C., Sakai, R., Sifrim, A., Aerts, J., Moreau, Y. and De Moor, B. (2014) 'A self-tuning genetic algorithm with applications in biomarker discovery', *IEEE 27th International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, pp.233–238.

Ray, S.S., Bandyopadhyay, S. and Pal, S.K. (2007) 'Genetic operators for combinatorial optimization in TSP and microarray gene ordering', *Applied Intelligence*, Vol. 26, No. 3, pp.183–195.

Reinelt, G. (2012) *Tsplib95, 1995* [online] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95 (accessed 14 October 2019).

Rey, A., Prieto, M., Gómez, J., Tenllado, C. and Hidalgo, J.I. (2018) 'ACPU-GPU parallel ant colony optimization solver for the vehicle routing problem', *International Conference on the Applications of Evolutionary Computation*, Springer, pp.653–667.

Schmitt, L.J. and Amini, M.M. (1998) 'Performance characteristics of alternative genetic algorithmic approaches to the traveling salesman problem using path representation: an empirical study', *European Journal of Operational Research*, Vol. 108, No. 3, pp.551–570.

Sombat, A., Saleewong, T. and Kumam, P. (2018) 'Perspectives and experiments of hybrid particle swarm optimization and genetic algorithms to solve optimization problems', *International Econometric Conference of Vietnam*, Springer, pp.290–297.

Somhom, S., Modares, A. and Enkawa, T. (1999) 'Competition-based neural network for the multiple travelling salesmen problem with minmax objective', *Computers & Operations Research*, Vol. 26, No. 4, pp.395–407.

Tajani, C., Abdoun, O. and Abouchabaka, J. (2017b) 'Optimization approach based on immigration strategies for symmetric traveling salesman problem', *Int. J. Open Problems Compt. Math.*, Vol. 10, No. 2, pp.38–53.

Tajani, C., Abdoun, O. and Lahjouji, A. (2017a) 'Genetic algorithm adopting immigration operator to solve the asymmetric traveling salesman problem', *International Journal of Pure and Applied Mathematics*, Vol. 115, No. 4, pp.801–812.

Tinós, R. and Yang, S. (2007) 'Genetic algorithms with self-organizing behaviour in dynamic environments', in *Evolutionary Computation in Dynamic and Uncertain Environments*, pp.105–127, Springer, Berlin, Heidelberg.

Toğan, V. and Daloğlu, A. T. (2008) 'An improved genetic algorithm with initial population strategy and self-adaptive member grouping', *Computers & Structures*, Vol. 86, No. 11, pp.1204–1218.

Van Hulle, M.M. (2012) 'Self-organizing maps', in *Handbook of Natural Computing*, pp.585–622, Springer-Verlag, Berlin, Heidelberg.

Wang, H., Wang, W., Cui, Z., Zhou, X., Zhao, J. and Li, Y. (2018) 'A new dynamic firefly algorithm for demand estimation of water resources', *Information Sciences*, Vol. 438, No. 1, pp.95–106.

Whitley, D., Hains, D. and Howe, A. (2009) 'Tunneling between optima: partition crossover for the traveling salesman problem', *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, ACM, pp.915–922.

Whitley, D., Hains, D. and Howe, A. (2010) 'A hybrid genetic algorithm for the travelling salesman problem using generalized partition crossover', *Parallel Problem Solving from Nature, PPSN XI*, pp.566–575.

Yang, X-S. (2010a) *Nature-Inspired Metaheuristic Algorithms*, pp.242–246, Luniver Press, Beckington, UK.

Yang, X-S. (2010b) 'A new metaheuristic bat-inspired algorithm', in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp.65–74.

Yang, X-S. and Deb, S. (2009) 'Cuckoo search via Lévy flights', *World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009*, IEEE, pp.210–214.

Yugay, O., Kim, I., Kim, B. and Ko, F.I. (2008) 'Hybrid genetic algorithm for solving traveling salesman problem with sorted population', *Third International Conference on Convergence and Hybrid Information Technology, 2008. ICCIT'08*, IEEE, Vol. 2, pp.1024–1028.

Zhou, X., Miao, F. and Ma, H. (2018) 'Genetic algorithm with an improved initial population technique for automatic clustering of low-dimensional data', *Information*, Vol. 9, No. 4, p.101.