Complex & Intelligent Systems (2020) 6:1–14
https://doi.org/10.1007/s40747-019-0102-7

ORIGINAL ARTICLE

Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator

Abid Hussain[1] · Yousaf Shad Muhammad[1]

Received: 27 November 2018 / Accepted: 15 March 2019 / Published online: 2 April 2019

Abstract

As an intelligent search optimization technique, genetic algorithm (GA) is an important approach for non-deterministic polynomial (NP-hard) and complex nature optimization problems. GA has some internal weakness such as premature convergence and low computation efficiency, etc. Improving the performance of GA is a vital topic for complex nature optimization problems. The selection operator is a crucial strategy in GA, because it has a vital role in exploring the new areas of the search space and converges the algorithm, as well. The fitness proportional selection scheme has essence exploitation and the linear rank selection is influenced by exploration. In this article, we proposed a new selection scheme which is the optimal combination of exploration and exploitation. This eliminates the fitness scaling issue and adjusts the selection pressure throughout the selection phase. The χ^2 goodness-of-fit test is used to measure the average accuracy, i.e., mean difference between the actual and expected number of offspring. A comparison of the performance of the proposed scheme along with some conventional selection procedures was made using TSPLIB instances. The application of this new operator gives much more effective results regarding the average and standard deviation values. In addition, a two-tailed t test is established and its values showed the significantly improved performance by the proposed scheme. Thus, the new operator is suitable and comparable to established selection for the problems related to traveling salesman problem using GA.

Keywords Genetic algorithm · Selection pressure · Selection operators · Statistical analysis · Traveling salesman problem

Introduction

Several modern meta-heuristic algorithms have been developed during the last five decades for solving the non-deterministic polynomial (NP-hard) and complex nature optimization problems. According to some specified criteria, these algorithms can be divided into different groups such as stochastic, deterministic, population, and iterative-based, etc. If an algorithm is trying to improve the solution according to the probabilistic rules, it is called stochastic algorithm. If an algorithm is trying to increase the solution quality with a set of solutions, it is called population-based and to seek the better solution to using multiple iterations called an iterative approach. The two important classifications of population-based algorithms are swarm intelligence and evolutionary approaches which depend on simulation theory with natural phenomenon.

Genetic algorithm (GA) is one of the most popular methods of evolutionary algorithms. It was first established on the theoretical basis by Holland [1]. GA is a universal optimization approach which relies on one of the most important criteria of Darwin's evolution process, as shown in Fig. 1. Usually, GA generates a better solution from all the possible solutions of a population based on the survival of the fittest principle. The random population of individuals with different encodes, such as binary, permutation, or real, etc., is created first. In nature, the most suitable individuals are likely to survive and mate. GA iteratively generates new chromosomes with the help of two operators, i.e., crossover and mutation. The process is repeated until or unless the required criteria such as convergence, a fixed time or a number of iterations are met. The objective is the solution with high astounding fitness values which are remarkable in the search process towards the optimal solution. The most attractive feature of GA is that it has the ability to explore the search

✉ Abid Hussain
abid0100@gmail.com

Yousaf Shad Muhammad
yousuf@qau.edu.pk

[1] Department of Statistics, Quaid-i-Azam University, Islamabad, Pakistan

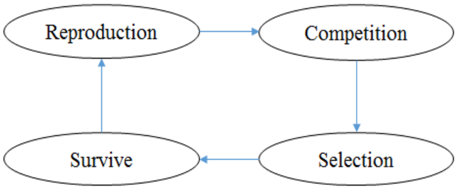

Fig. 1 Darwin's evolution principle

space with the help of the entire population of individuals [2]. Recently, Bayesian network used as an adaptive parameter setting tool to enhance the performance of GA for multimodal problems [3]. A lot of work and applications have been highlighted about GAs by Goldberg [4].

A very common issue about GA is premature convergence to find the optimal solution of a problem. This is strongly linked to the loss of population diversity. If it is very low then a very quick convergence will be observed by GA; otherwise, time-consuming and may cause wastage of computational resources. Hence, there is essential to find a trade-off between exploration (i.e., exploring the new areas of search space) and exploitation (i.e., using already detected points to search the optimum). Therefore, the performance of the GA highly depends on its genetic operators, in general. The first operator is selection being used to choose the set of chromosomes for mating process, the crossover is the second one and used to create new individuals, and the last one is the mutation used for random changes. The balance between exploration and exploitation can be adjusted either by selection pressure in a selection approach or by the recombination operators with adjustment of their probabilities.

The selection scheme is the procedure to choose a subpopulation (set of individuals) from the current population that will form the next population. GA is one of those algorithms whose performance is highly affected by the choice of selection operator. Without this mechanism, GA is only simple random sampling giving different results in each generation. Hence, we can say that the selection operator is the backbone of the GA process. Usually, the choice of the selection mechanism depends on the complexity of the problem. A hard approach combined with a conservative replacement mechanism and soft one manipulate an algorithm without sufficient exploring capability which may cause to stuck off on local optima.

There are several selection operators used and reviewed in the literature. A study about various selection approaches and results showed that different schemes perform well in different problems [5]. Thus, the most suitable selection approach has to be chosen in relation to a specific problem to enhance the optimality of desired result. Goldberg and Deb [6] did a comprehensive study of some traditional selection methods through the solutions of differential equations. Another popular study to adjust the probabilistic noise level throughout the mating pool to regulate the selection pressure [7]. Abd-Rahman et al. [8] established a hybrid roulette-tournament selection operator for solving a real-valued shrimp diet formulation problem which can also be generalized to evolutionary algorithm-related problems. A detailed study is about the selection process in GA and examined some common issues in various selection operators in Ref. [5].

The main objective of this study is to present the performance of selection operators that have a major impact on the GAs process. In this way, a new selection operator is proposed that intended to enhance the average quality of the population and gives a better trade-off between exploration and exploitation.

The rest of this article is presented as follows: in "Background" we present the background of selection schemes. The proposed selection operator is presented in "Proposed selection operator" with the statistical properties of a sampling algorithm. The traveling salesman problem (TSP) is discussed and reviewed in "Test problem (traveling salesman problem)". Performance evaluation of the proposed scheme and conclusions are given in "Performance evaluation" and "Conclusions", respectively.

Background

The first selection mechanism for GA was fitness proportional selection (FPS), which was introduced by Holland [1]. Now, it has become the most prevalent selection approach which used the concept of proportionality. It works as the fitness value of each individual in a population corresponds to the area of roulette wheel proportions. Then, an individual is marked by the roulette wheel pointer after it has spun. This operator gives individuals, a probability p_i of being selected Eq. (1) that is directly proportionate to their fitness:

$$p_i = \frac{f_i}{\sum_{j=1}^{K} f_j}; \quad i \in \{1, 2, \ldots, K\}, \tag{1}$$

where K is the size of the population and f_i is the value of fitness function for the individual i. Thus, individuals who have better fitness values may have a higher chance of being selected as parents.

The FPS has been widely used selection scheme in various fields such as spanning tree [9], scheduling [10,11], sources allocation problem [12], menu planning [13], and traveling salesman problem [14]. Throughout the selection procedure, there is no change in the segment size and selection probability. It is easy to implement and gives a high probability for the best individual; these aspects are the main strengths [15]. Another advantage of this approach is that it provides no bias with unlimited spread [16]. However, the difficulty is encountered when a significant difference appears in the fitness values [14,17,18]. The scaling problem which is the major drawback of this scheme was first pointed out by Grefenstette [19]. It has happened when population evolves, the ratio between the variance and the fitness average becomes increasingly small. The selection pressure, therefore, drops as the population converges [7]. On the other hand, high selection pressure may lead to premature convergence to a sub-optimal solution.

In the literature, there are some alternative techniques to overcome the above drawbacks. The most popular technique is the linear rank selection (LRS) scheme proposed by Baker [20]. It sorts the individuals in the sequence as worst to best according to the fitness and allocates them a survival probability proportional to their rank order. After this task, a sampling procedure (i.e., roulette wheel sampling) is used to select the individuals for mating process. In this way, the LRS can maintain a constant selection pressure throughout in the sampling process, because it introduces a uniform scaling across the population. Therefore, a unique selection probability is always assigned to the best individual, regardless of its fitness value. Another advantage of the LRS is that it behaves in a more robust manner than other techniques. The selection probability of an individual through this scheme is assigned according to the following formula:

$$p_i = \frac{1}{K}\left(\eta^- + (\eta^+ - \eta^-)\frac{i-1}{K-1}\right); \quad i \in \{1, 2, \ldots, K\}. \tag{2}$$

Here, $\frac{\eta^-}{K}$ and $\frac{\eta^+}{K}$ are the probabilities of worst and best chromosomes to be selected, respectively. All the individuals get a different rank even if they have the same fitness value. The conditions $\eta^+ = 2 - \eta^-$ and $\eta^- \geq 0$ must be fulfilled. The selective pressure can be adjusted by varying η^+, the parametric value. As remarked by Baker, if $\eta^+ = 2$, then all individuals would be within 10% of the mean and the population is driven to convergence during every generation. Baker recommended value of $\eta^+ = 1.1$ to control the selection pressure. The weakness of this scheme is that it can lead to slower convergence, because there is no significant difference between the best and other individuals. The selection probability of two consecutive chromosomes by the same amount is regardless of whether the gap between their fitness is larger or smaller [7].

Another rank-based selection scheme is exponential ranking selection (ERS). It works similar as to LRS, except for the non-linear assignment of probabilities to the individuals. A constant ratio r is used to assign the rank-based values to the individuals, such that $0 < r < 1.0$ ($r \simeq 1.0$). The selection probability for the ith ranked individual through this scheme is assigned according to the following formula:

$$p_i = \frac{r^{K-i}(1-r)}{1-r^K}; \quad i \in \{1, 2, \ldots, K\}. \tag{3}$$

The tournament selection (TS) is also widely used as an alternative to FPS. In TS, first, randomly select the t (where t is the predefined tournament size) individuals from the population and then they compete against each other based on their fitness. An individual with higher fitness value is declared as a winner and selected for mating process. The selection pressure can be adjusted with change the tournament size [7]. Usually, the most used tournament size is 2 (binary tournament selection (BTS)), which is the simplest form of TS [21]. However, the larger tournament size can be used to enhance the competition among individuals, but it leads to loss of population diversity [22,23]. As shown by Back [24], an individual i is selected for t tournament using Eq. (4), where K is the size of the population. If the tournament size equals the population size, then the TS will be the approximate to the deterministic selection procedure [7]:

$$p_i = \frac{1}{K^t}((i)^t - (i-1)^t); \quad i \in \{1, 2, \ldots, K\}. \tag{4}$$

Another case of the TS is probabilistic two-tournament selection (PTS) was presented by Julstrom [25]. The two individuals without replacement are chosen at random from the population and the winner of this tournament will be selected for mating process with a probability q, such that $0.5 < q < 1.0$. In this scheme, the loser can also be selected for mating process with the probability $(1 - q)$. Thus, the selection probability of an individual through this scheme is assigned according to the following rule:

$$p_i = \frac{2(i-1)}{K(K-1)}q + \frac{2(K-i)}{K(K-1)}(1-q); \quad i \in \{1, 2, \ldots, K\}. \tag{5}$$

Moreover, throughout the evolution process, a fixed and a suitable adjustment of the selection pressure is a difficult task. An ideal situation may exist, if the selection pressure is low at the early stage of the search to gives a free hand to an exploration of the solution space and enhance at the ending stage to help the algorithm for convergence [26]. Hence, to trade-off between these two competing criteria, an adjustable selection pressure must desired [7]. The main contribution of this article is in the development of the proposed selection approach which reduces the weakness associated with FPS and LRS in the GA procedure. The proposed approach is based on the ranking scheme which splits the individuals after ranking and then assign them probabilities for selection. This will increase the competition among individuals to be selected for mating process to regulate the selection pressure. The detail is given in the next section.

Proposed selection operator

Motivation

The LRS has a small range of selection pressures (i.e., for a population of K individuals, the selection probability of the fittest individual is fallen must be between $\frac{1}{K}$ and $\frac{2}{K}$). The LRS introduces slow convergence speed and sometimes converges to a sub-optimal solution as less fit individuals may be preserved from one generation to another. In GA, the FPS has the essence of exploitation, while LRS is influenced by exploration. The information about the relative evaluation of individuals is ignored, all cases are treated uniformly regardless of the magnitude of the problem and, finally, the schema theorem is violated. LRS prevents too quick convergence and differs from FPS in terms of selection pressure. This discussion suggests that, whenever a selection procedure is used, some kind of adaptation of the selection pressure is highly desirable.

Split rank selection

In this research, we propose an alternative selection scheme [split rank selection (SRS)] that maintains a fine balance between exploration and exploitation. This approach not only eliminates the fitness scaling problem, but also provides an adequate selection pressure throughout the selection phase. In this scheme, all individuals are sorted from worst to best according to their fitness values. All the individuals get a different rank even if they have the same fitness value. Consider a population, a combination of K individuals, i.e., population size (usually it is even). Now, we divide the K individuals into two equal portions.

The top half portion is considering from individual 1 to individual $\frac{K}{2}$. The result (sum up) of this portion of the series is as follows:

$$\frac{K}{4}\left(\frac{K}{2}+1\right). \tag{6}$$

Now, sum up the last half portion of the series, i.e., from $\frac{K}{2}+1$ to K is

$$\frac{K^2}{4}+\frac{K}{4}\left(\frac{K}{2}+1\right). \tag{7}$$

Hence, the probability distribution according to the individual's rank as

$$p(i)=\begin{cases}\lambda^-\left(\frac{8i}{K(K+2)}\right); & i \le \frac{K}{2}\\ \lambda^+\left(\frac{8i}{K(3K+2)}\right); & i > \frac{K}{2},\end{cases} \tag{8}$$

where $\lambda^- + \lambda^+ = 1$ and $\lambda^- \ge 0$ must be satisfied. The selection pressure can be restrained by varying λ^+, the tuning parameter, in the selection phase. To maintain balance of exploitation and exploration, we adjust this parameter $\lambda^+ = 0.7$, i.e., 70% portion is assigned to the last half individuals. This gives

$$p(i)=\begin{cases}\frac{12i}{5K(K+2)}; & i \le \frac{K}{2}\\ \frac{28i}{5K(3K+2)}; & i > \frac{K}{2}.\end{cases} \tag{9}$$

We also derive the formula to select individuals if the population size is odd (usually this happens rarely in literature). First of all, we divide the K individuals in $\frac{K-1}{2}$ and $\frac{K+1}{2}$ portions. The top portion is considering from individual 1 to individual $\frac{K-1}{2}$. The result (sum up) of this portion of the series is as follows:

$$\frac{K^2-1}{8}. \tag{10}$$

Now, sum up the last portion of the series, i.e., from $\frac{K+1}{2}$ to K is

$$\frac{K+1}{2}+\frac{K+3}{2}+\frac{K+5}{2}+\cdots+\frac{K+K}{2} \tag{11}$$

$$\frac{K}{4}(K+1)+\frac{1}{8}(K+1)^2 \tag{12}$$

$$\frac{(K+1)(3K+1)}{8}. \tag{13}$$

Hence, the probability distribution according to the individual's rank is as follows:

$$p(i)=\begin{cases}\lambda^-\left(\frac{8i}{(K^2-1)}\right); & i \le \frac{K-1}{2}\\ \lambda^+\left(\frac{8i}{(K+1)(3K+1)}\right); & i > \frac{K-1}{2}.\end{cases} \tag{14}$$

We compare the proposed operator with LRS, ERS, BTS, and PTS for 150 individuals at various parameters and depicted in Fig. 2. We used the most optimal parametric values from the literature to achieve a maximal performance from the said operators. For example, Baker shows that LRS performs best at $\eta^+ = 1.1$. For tournament selection, binary is much better size for selection; otherwise, it may lead to loss of diversity. For ERS, the parametric value 'r' is very close to 1 for better performance and we use $r = 0.99$. For PTS, the parametric value 'q' is allowed to be within $0.5 - 1$, but a high value is recommended to control population diversity so we take $q = 0.8$. We see that the SRS works efficiently at $\lambda^+ = 0.7$ for bad individuals and gives slightly better probability than BTS but not as much as given to them by LRS.

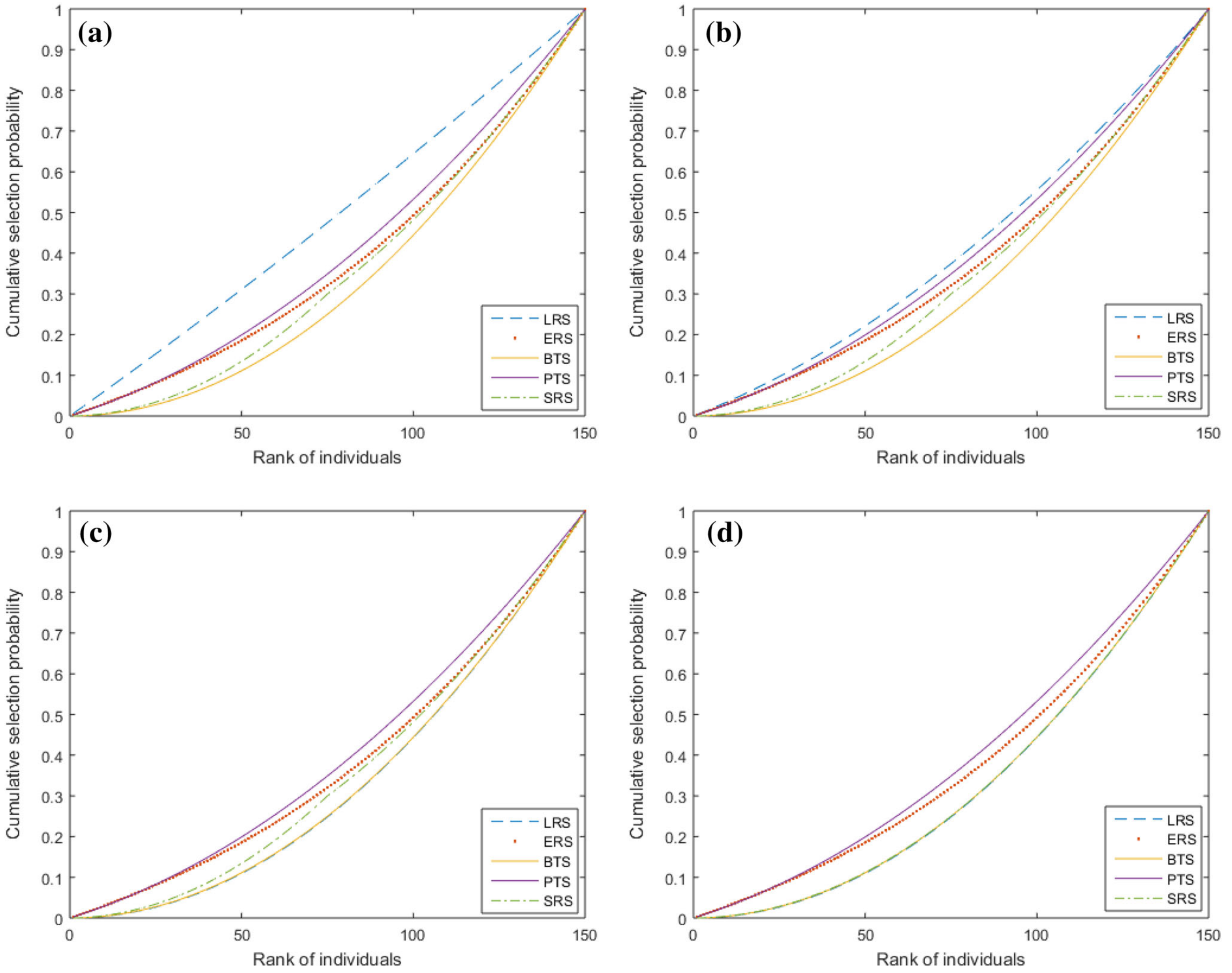

Fig. 2 Comparison among LRS, ERS, BTS, PTS, and SRS for 150 individuals, **a** $\eta^+ = 1.1$, $r = 0.99$, $t = 2$, $q = 0.8$, and $\lambda^+ = 0.7$; **b** $\eta^+ = 1.5$, $r = 0.99$, $t = 2$, $q = 0.8$ and $\lambda^+ = 0.7$; **c** $\eta^+ = 2$, $r = 0.99$, $t = 2$, $q = 0.8$ and $\lambda^+ = 0.7$; **d** $\eta^+ = 2$, $r = 0.99$, $t = 2$, $q = 0.8$ and $\lambda^+ = 0.75$

The sampling procedure

In a two-step selection procedures, i.e., FPS, LRS, ERS, and SRS, etc., a sampling mechanism is required to choose the individuals for mating process. That sampling mechanism fills the mating pool with the individual's copies of the given population, while respecting the selection probabilities p_i, such that the expected and observed number of individuals are equals. Among the broad variety of sampling mechanisms, we used roulette wheel sampling scheme (or Monte Carlo sampling) for testing the accuracy of the proposed SRS operator.

The χ^2 goodness-of-fit measure

To measure the average difference between the expected and actual numbers of offspring, the χ^2 measure, as a tool for the average accuracy was first introduced by Schell et al. [27]. At first, there are c disjoint classes as $\{C_1, C_2, \ldots, C_c\}$ where $C_j \subset \{1, 2, \ldots, K\}$ and $\cup_{j=1}^{c} C_j = \{1, 2, \ldots, N\}$. Let $\xi_j = \sum_{i \in C_j} e_i$ denotes the overall expectation and $O_j = \sum_{i \in C_j} o_i$ is for the observed (actual) copies of individuals in mating pool after the sampling procedure. Ideally, ξ_j should be of the order K/c for $1 \leq j \leq c$, so that each class contains the same individuals on average and it should be at least 10 to obtain the required stochastic accuracy. Schell et al. [27] defined the Chi-square test as a measure to determine the accuracy of the sampling process as follows:

$$\chi := \sum_{j=i}^{c} \frac{(\xi_j - O_j)^2}{\xi_j}. \tag{15}$$

In the roulette wheel sampling situation, the aforementioned constraint (i.e., $\xi_j \geq 10$), however, χ should follow the Chi-square distribution with $c-1$ degrees of freedom, since this is the asymptotic distribution of χ under multinomial distributed o_i when $K \rightarrow \infty$. In this research, we present the results for a fixed parameter setting, i.e., $\lambda^+ = 0.7$, the size of population $K = 150$, classes $c = 10$, and total number of tests $s = 150$.

Table 1 presents the probability distribution of SRS and the corresponding overall expectation that are very close to 150/10. We used $\chi^{S,R}$ to measure the results of χ. In $\chi^{S,R}$, S represents the proposed operator that assigns the probabilities to individuals and R is a type of sampling algorithm, i.e., roulette wheel. The main objective of this test is to estimate the expectation and variance. The population is randomly generated with predefined fixed individuals, and used the probability distribution R to assign them probabilities for selection and after that the sampling scheme R is applied to obtain instances of o_i, O_j and $\chi^{S,R}$ respectively. From the sequence $(\chi_k^{S,R})_{1 \le k \le s}$, the sample mean and variance can be calculated as follows:

$$\hat{e}^{(S,R)} = \frac{1}{s}\sum_{k=1}^{s}\chi_k^{S,R} \tag{16}$$

$$\hat{\sigma}^{2(S,R)} = \frac{1}{s-1}\sum_{k=1}^{s}(\chi_k^{S,R} - \hat{e}^{(S,R)})^2. \tag{17}$$

This scheme is compared to the theoretical χ^2_{c-1} distribution at 99% confidence interval. For 10 classes, the mean and variance of Chi-square are $c-1 = 9$ and $2(c-1) = 18$. The corresponding estimates of $\hat{e}$ and $\hat{\sigma}^2$ are 8.5739 and 19.6010, respectively. These estimates are almost the same and a more symbolic representation of a comparison of accuracy between assigning the probabilities to individuals and the number of copies corresponding to their probabilities come in the mating pool. The empirical results confirm the average behavior of the sampling scheme with respect to the probability distribution of SRS. The roulette wheel sampling provides the empirical distribution function that can not be significant from the theoretical χ^2_{c-1} distribution by $\hat{e}$ and $\hat{\sigma}^2$ statistics.

Table 1 Classes C_j and overall expectation ξ_j for SRS

j	C_j	ξ_j
1	1–43	14.9368
2	44–61	14.9211
3	62–75	15.1421
4	76–90	15.4248
5	91–103	15.6230
6	104–114	14.8549
7	115–124	14.8053
8	125–133	14.3841
9	134–142	15.3876
10	143–150	14.5203

Test problem (traveling salesman problem)

The traveling salesman problem (TSP) is one of the most famous benchmark, significant, and historic hard combinatorial optimization problems. The main objective of TSP is to find the shortest Hamiltonian tour in a complete graph with n nodes. It was documented by Euler in 1759 (his interest was how to get rid of the knight's tour problem) [28]. It is a fundamental problem in the fields of computer science, engineering, operations research, discrete mathematics, and graph theory. In this problem, a salesman visits all cities (nodes) exactly once (the constraint) and then returns to the initial point to complete a tour. It has many applications such as a variety of vehicle routing [29], scheduling [30], and bioinformatics [31] which can easily be transformed into the TSP.

If there are 'n' cities, a distance matrix $C = [c_{ij}]_{n\times n}$ is searched for a permutation $\lambda : \{0, \ldots, n-1\} \longrightarrow \{0, \ldots, n-1\}$, where c_{ij} is the distance from city i to city j, which minimizes the traveled distance, $f(\lambda, C)$:

$$f(\lambda, C) = \sum_{i=0}^{n-1} d(c_{\lambda(i)}, c_{\lambda(i+1)}) + d(c_{\lambda(n)}, c_{\lambda(1)}), \tag{18}$$

where $\lambda(i)$ represents the location of city i in each tour, $d(c_i, c_j)$ is the distance between city i to city j and (x_i, x_j) is a specified position of each city in a tour in the plane, and the Euclidean distances of the distance matrix C between the city i and j is expressed as follows:

$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{19}$$

TSP is easy to understand but very difficult to solve, i.e., for '100' cities, there are 10^{155} possible ways to find the tour. This is the reason to say that TSP is a non-deterministic polynomial (NP-hard) problem [32,33]. These type of problems cannot be solved using the traditional optimization approaches like gradient-based methods. To achieve the optimal solution within a considerable amount of time, heuristic approaches are efficient at handling the NP-hard problems [34–37]. The GA has also been used to solve this problem in several aspects [28,38–44].

Performance evaluation

In this section, we evaluate the performance of the SRS in comparison to other selection schemes. At first, we present basic information about benchmarks and the parameters setting for GA in "Computational testing methodology". Second, MATLAB software (version R2015a) was used to compare the simulation study among selection operators, and a detailed discussion on results is given in "Simulation results and discussion".

Computational testing methodology

In this research, computational experiments on ten different instances taken from the library of traveling salesman problem (TSPLIB) [45] are solved to compare the proposed scheme along with five competing selection methods. The test benchmarks are Euclidean, two-dimensional symmetric and asymmetric problems within 34–561 cities and given in Table 2. In addition, we consider the three most widely used crossover schemes, namely order crossover (OX), partially mapped crossover (PMX) and cycle crossover (CX). The exchange mutation (EM) as a mutation operator is used throughout our simulation study. These are the state-of-the-art genetic operators and a detailed discussion is given in Ref. [28]. Therefore, there are three group of experiments conducted as six selection schemes examined with each of three crossovers and one mutation operators. Table 3 presents the genetic operators and parameter values which are under consideration in our simulation study.

Since GA belongs to the class of probabilistic search algorithms, we use the two-sampled t test as a statistical hypothesis testing [46]. The experiments were performed in 30 independent trials (each pair of $n_1 = n_2 = 30$) for each instance to achieve a comparable solution. The two-tailed t test values are calculated using Eq. (20); where $\bar{x_1}$ and s_1 are, respectively, the average and standard deviation (SD) of SRS and $\bar{x_2}$ and s_2 are, respectively, the average and SD of other competitor operators (i.e. FPS, LRS, ERS, BTS, and PTS). In this study, we set our null hypothesis in the following way 'SRS convergences at least as fast as other selection operators in comparison'. Throughout this study, all the statistical differences are shown at $p = 0.05$ (95% confidence) level of significance using the two-sample (independent samples) t test with 58 degrees of freedom. The two-tailed t test value indicates whether a significant improvement by SRS ($t \leq -2.00$) or significant degradation by SRS ($t \geq 2.00$). But within the range ($-2.00 < t < 2.00$) of two-tailed t test score does not reflect the reasonable statistical evidence to confirm or refute our null hypothesis, which indicates that there is no statistical significance between the two approaches:

$$t = \frac{\bar{x_1} - \bar{x_2}}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \tag{20}$$

where

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}.$$

Table 2 The benchmark problems

Problem name	No. of cities	Optimal tour length
ftv33	34	1286
berlin52	52	7542
ft70	70	38,673
kroA100	100	21,282
ftv170	171	2755
brg180	180	1950
pr226	226	80,369
rbg323	323	1326
rbg403	403	2465
pa561	561	2763

Table 3 Parametric configuration for GA

Parameter	Setting
Representation	Permutation
Population size	150
Crossover criteria	PMX, OX, and CX
Crossover rate	80%
Mutation method	EM
Mutation rate	5%
Maximum generation	5000
Number of trails	30
Replacement in GA	Steady-state GA

Simulation results and discussion

Table 4 summarizes the results of six competing selection schemes with PMX and EM as crossover and mutation operators respectively. Results compare on the basis of average, SD, and improved performance of the SRS in percentage (%) values. The significant improvements in the results of SRS with respect to each other approach are indicated through t values. The proposed operator is indicated less average values for all ten benchmarks with stable results (low SD), as well. According to the critical value ($t = -2.00$), all computed t scores are less than -2.00 for all ten benchmark instances and bold t test values have shown the significantly improved performance by the proposed operator. The other t test values which are not bold (non-significant), but negative values indicates a slightly improved performance with respect to an average by the proposed operator. In other words, the simulation results found by the SRS are statistically significant

Table 4 Results of different selection strategies with PMX (crossover) and EM (mutation) operators

Instance	Optimal	Selection scheme	Average	Improvement in SRS (%)	SD	*t* test
ftv33	1286	FPS	1480	5.95	117	− **3.29**
		LRS	1503	7.39	138	− **3.71**
		ERS	1588	12.34	223	− **4.48**
		BTS	1461	4.72	110	− **2.68**
		PTS	1551	10.25	186	− **4.23**
		SRS	**1392**	–	**88**	–
berlin52	7542	FPS	7703	1.17	187	− **2.28**
		LRS	7717	1.35	173	− **2.79**
		ERS	7903	3.67	235	− **6.13**
		BTS	7695	1.07	148	− **2.44**
		PTS	7812	2.55	199	− **4.80**
		SRS	**7613**	–	**109**	–
ft70	38,673	FPS	40,692	2.04	1333	− **2.97**
		LRS	40,174	0.77	1086	− 1.29
		ERS	42,239	5.63	1457	− **7.95**
		BTS	39,954	0.22	958	− 0.41
		PTS	40,578	1.76	1223	− **2.73**
		SRS	**39,863**	–	**747**	–
kroA100	21,282	FPS	21,883	1.43	418	− **2.98**
		LRS	21,962	1.78	535	− **3.23**
		ERS	22,808	5.42	876	− **7.06**
		BTS	21,806	1.08	506	− **2.01**
		PTS	21,980	1.90	435	− **3.83**
		SRS	**21,571**	–	**392**	–
ftv170	2755	FPS	3086	4.18	249	− **2.40**
		LRS	3129	5.50	281	− **2.93**
		ERS	3266	9.46	301	− **4.96**
		BTS	3163	6.51	266	− **3.65**
		PTS	3178	6.95	283	− **3.74**
		SRS	**2957**	–	**157**	–
brg180	1950	FPS	2199	6.18	217	− **2.74**
		LRS	2241	7.94	239	− **3.36**
		ERS	2189	5.76	226	− **2.47**
		BTS	2118	2.60	211	− 1.13
		PTS	2254	8.47	261	− **3.39**
		SRS	**2063**	–	**164**	–
pr226	80,369	FPS	82,180	0.73	1392	− **2.02**
		LRS	82,321	0.93	1444	− **2.42**
		ERS	83,233	1.99	1554	− **5.10**
		BTS	82,115	0.66	1101	− **2.11**
		PTS	82,821	1.50	1327	− **4.30**
		SRS	**81,577**	–	**863**	–
rbg323	1326	FPS	1631	7.48	218	− **2.24**
		LRS	1694	10.92	262	− **3.06**
		ERS	1659	9.04	236	− **2.64**
		BTS	1621	6.91	209	− **2.11**

Table 4 continued

Instance	Optimal	Selection scheme	Average	Improvement in SRS (%)	SD	*t* test
		PTS	1712	11.86	241	− **3.53**
		SRS	**1509**	–	**203**	–
rbg403	2465	FPS	2844	3.97	247	− **2.01**
		LRS	2803	2.57	275	− 1.19
		ERS	2897	5.73	271	− **2.77**
		BTS	2788	2.04	234	− 1.05
		PTS	2820	3.16	260	− 1.52
		SRS	**2731**	–	**185**	–
pa561	2763	FPS	2979	3.52	191	− **2.66**
		LRS	2958	2.84	165	− **2.37**
		ERS	3024	4.96	186	− **3.87**
		BTS	2911	1.27	128	− 1.24
		PTS	2993	3.98	179	− **3.16**
		SRS	**2874**	–	**102**	–

Table 5 Results of different selection strategies with OX (crossover) and EM (mutation) operators

Instance	Optimal	Selection scheme	Average	Improvement in SRS (%)	SD	*t* test
ftv33	1286	FPS	1498	8.74	183	− **3.50**
		LRS	1471	7.07	141	− **3.39**
		ERS	1532	10.77	196	− **4.18**
		BTS	1424	4.00	122	− **2.05**
		PTS	1510	9.47	157	− **4.32**
		SRS	**1367**	–	**91**	–
berlin52	7542	FPS	7671	1.15	172	− **2.20**
		LRS	7618	0.46	155	− 0.93
		ERS	7809	2.89	201	− **5.11**
		BTS	7607	0.32	158	− 0.63
		PTS	7732	1.93	167	− **3.80**
		SRS	**7583**	–	**135**	–
ft70	38,673	FPS	40,468	2.60	1106	− **3.92**
		LRS	39,993	1.44	1215	− **2.03**
		ERS	40,897	3.62	1414	− **4.73**
		BTS	39,641	0.57	1223	− 0.79
		PTS	40,376	2.38	1288	− **3.26**
		SRS	**39,417**	–	**968**	–
kroA100	21,282	FPS	21,598	1.26	392	− **3.43**
		LRS	21,593	1.24	409	− **3.25**
		ERS	21,879	2.53	513	− **5.55**
		BTS	21,487	0.75	388	− **2.05**
		PTS	22,036	3.22	469	− **7.71**
		SRS	**21,326**	–	**186**	–
ftv170	2755	FPS	3164	5.91	251	− **3.52**
		LRS	3107	4.18	196	− **2.90**
		ERS	3223	7.63	277	− **4.29**
		BTS	3083	3.44	156	− **2.70**

Table 5 continued

Instance	Optimal	Selection scheme	Average	Improvement in SRS (%)	SD	*t* test
		PTS	3198	6.91	188	− **5.06**
		SRS	**2977**	–	**148**	–
brg180	1950	FPS	2140	5.79	229	− **2.44**
		LRS	2177	7.40	212	− **3.34**
		ERS	2246	10.24	258	− **4.16**
		BTS	2053	1.80	193	− 0.81
		PTS	2217	9.07	207	− **4.23**
		SRS	**2016**	–	**158**	–
pr226	80,369	FPS	81,715	0.74	1012	− **2.62**
		LRS	81,967	1.05	1378	− **2.99**
		ERS	83,030	2.31	1554	− **6.09**
		BTS	81,878	0.94	1229	− **2.92**
		PTS	82,372	1.53	1331	− **4.52**
		SRS	**81,110**	–	**756**	–
rbg323	1326	FPS	1625	8.43	206	− **2.66**
		LRS	1666	10.68	215	− **3.37**
		ERS	1764	15.65	249	− **4.80**
		BTS	1597	6.83	223	− **2.02**
		PTS	1659	10.31	216	− **3.23**
		SRS	**1488**	–	**193**	–
rbg403	2465	FPS	2859	4.72	311	− **2.00**
		LRS	2776	1.87	268	− 0.85
		ERS	2864	4.89	318	− **2.04**
		BTS	2727	0.11	254	− 0.05
		PTS	2863	4.86	301	− **2.10**
		SRS	**2724**	–	**201**	–
pa561	2763	FPS	2955	3.25	114	− **3.51**
		LRS	2981	4.09	143	− **3.87**
		ERS	3107	7.98	168	− **7.00**
		BTS	2890	1.07	138	− 1.01
		PTS	3036	5.83	155	− **5.30**
		SRS	**2859**	–	**97**	–

and better than the other five selection approaches (i.e., FPS, LRS, ERS, BTS, and PTS).

The order crossover (OX) is used instead of PMX and simulation results are summarized in Table 5 for various selection operators. These results are also compared on the basis of average, SD and improved performance of the SRS in percentage (%) values. The *t* test is also used to measures not only improved but significant performance by the proposed SRS. The simulated results show less average values by SRS for all the benchmarks with consistent results (low SD), as well. According to the critical value ($t = -2.00$), all computed *t* scores are less than -2.00 for all ten benchmark instances and bold *t* test values have shown the significantly improved performance by the proposed operator. The other *t* test values which are not bold (non-significant), but negative values indicates a slightly improved performance with respect to an average by the proposed operator. The table shows that there is no positive *t* test value which means that no other operator is better than the proposed one in any case. Based on the simulation results, we can say that the proposed operator (SRS) is statistically significant and better than the other five selection approaches (i.e., FPS, LRS, ERS, BTS, and PTS).

We continue our simulation study to check the performance of the proposed operator along with other selection methods and different techniques of crossover and mutation operators. Likewise, in Table 6, we tested the performance of SRS with the pair of CX (crossover operator) and EM

Table 6 Results of different selection strategies with CX (crossover) and EM (mutation) operators

Instance	Optimal	Selection scheme	Average	Improvement in SRS (%)	SD	*t* test
ftv33	1286	FPS	1547	8.40	209	− **2.62**
		LRS	1562	9.28	218	− **2.85**
		ERS	1643	13.76	241	− **4.17**
		BTS	1509	6.10	225	− 1.78
		PTS	1596	11.22	257	− **3.16**
		SRS	**1417**	–	**173**	–
berlin52	7542	FPS	7723	1.19	199	− **2.02**
		LRS	7739	1.38	164	− **2.67**
		ERS	7951	4.01	243	− **6.16**
		BTS	7694	0.81	171	− 1.51
		PTS	7804	2.20	187	− **3.97**
		SRS	**7632**	–	**146**	–
ft70	38,673	FPS	40,731	2.07	1325	− **2.69**
		LRS	40,232	0.86	1277	− 1.12
		ERS	41,146	3.06	1362	− **3.95**
		BTS	40,128	0.60	1264	− 0.79
		PTS	41,089	2.93	1147	− **4.16**
		SRS	**39,887**	–	**1093**	–
kroA100	21,282	FPS	21,975	2.68	512	− **3.65**
		LRS	22,032	2.93	638	− **3.50**
		ERS	23,143	6.72	756	− **10.59**
		BTS	21,876	2.68	407	− **3.21**
		PTS	22,435	3.78	564	− **7.40**
		SRS	**21,588**	–	**274**	–
ftv170	2755	FPS	3190	5.80	276	− **3.16**
		LRS	3156	4.78	219	− **3.02**
		ERS	3256	7.71	325	− **3.78**
		BTS	3108	3.31	222	− **2.04**
		PTS	3269	8.08	297	− **4.26**
		SRS	**3005**	–	**164**	–
brg180	1950	FPS	2220	7.48	228	− **3.14**
		LRS	2215	7.27	201	− **3.28**
		ERS	2285	10.11	246	− **4.16**
		BTS	2188	6.12	225	− **2.55**
		PTS	2243	8.43	218	− **3.67**
		SRS	**2054**	–	**179**	–
pr226	80,369	FPS	82,997	1.60	1367	− **4.37**
		LRS	82,442	0.93	1056	− **2.98**
		ERS	83,564	2.26	1431	− **6.05**
		BTS	82,592	1.11	1108	− **3.46**
		PTS	83,670	2.39	1042	− **7.79**
		SRS	**81,673**	–	**941**	–
rbg323	1326	FPS	1723	10.50	226	− **3.23**
		LRS	1706	9.61	252	− **2.75**
		ERS	1859	17.05	320	− **4.55**
		BTS	1684	8.43	231	− **2.50**

Table 6 continued

Instance	Optimal	Selection scheme	Average	Improvement in SRS (%)	SD	*t* test
		PTS	1744	11.58	228	− **3.58**
		SRS	**1542**	–	**208**	–
rbg403	2465	FPS	2911	5.50	339	− **2.20**
		LRS	2827	2.69	297	− 1.15
		ERS	2943	6.52	364	− **2.51**
		BTS	2784	1.19	288	− 0.51
		PTS	2832	2.86	303	− 1.21
		SRS	**2751**	–	**209**	–
pa561	2763	FPS	3015	4.28	192	− **3.17**
		LRS	2997	3.70	178	− **2.88**
		ERS	3159	8.64	247	− **5.51**
		BTS	2955	2.33	164	− 1.90
		PTS	3068	5.93	156	− **5.18**
		SRS	**2886**	–	**113**	–

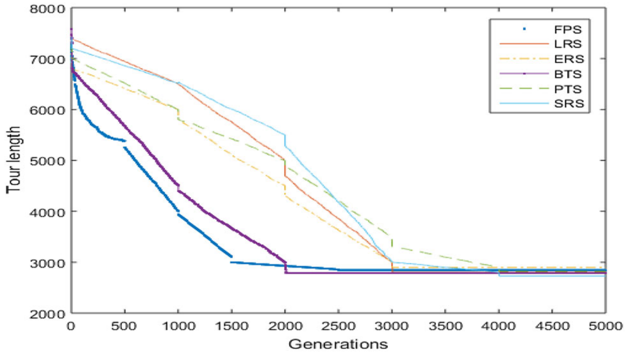

Fig. 3 Convergence of GA using PMX and EM for the instance ‘rbg403’

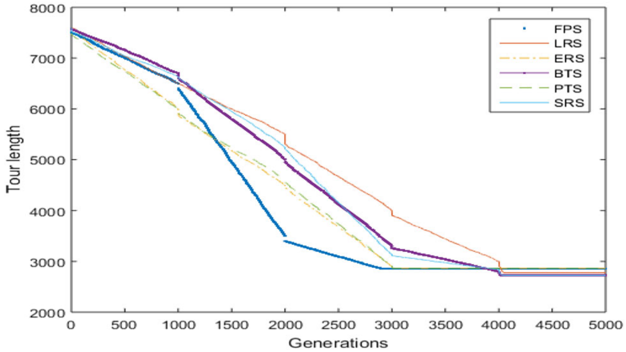

Fig. 4 Convergence of GA using OX and EM for the instance ‘rbg403’

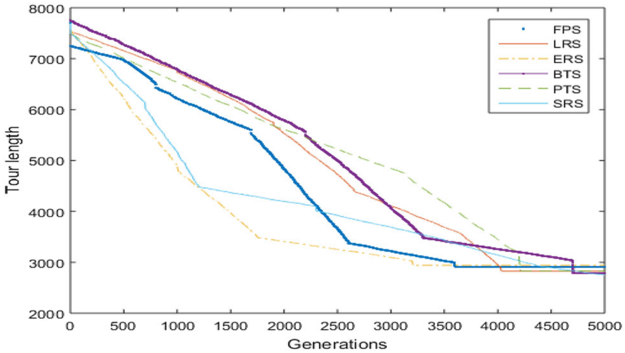

Fig. 5 Convergence of GA using CX and EM for the instance ‘rbg403’

(mutation operator). The simulation results indicate the lower average and SD values for all benchmarks by the SRS. Based on statistical perspectives, the SRS outperforms (bold *t* test values) all the other selection methods for all ten benchmark instances ($t \le -2.00$), but, in some cases, only BTS and LRS give non-significant results with the proposed operator. The non-bold *t* test values are all negative which means that the proposed operator is not worse than any other competing selection operators used in this study. Besides, we can clearly see from Figs. 3, 4 and 5 and analyses performed on the ‘rbg403’ instance that SRS produces lower average results using three different crossover and one mutation operators. We also observe that FPS and BTS produced faster results in early stages, but lead to premature convergence because of high selection pressure. On the other hand, the proposed operator work efficiently throughout the generations taking care of selection pressure and population diversity.

Conclusions

Exploration and exploitation are the two main techniques which employed normally to all the optimization methods. The fitness proportional selection approach has essence exploitation and linear rank approach is influenced by exploration. This article presented a new split ranked selection operator which is a great trade-off between exploration and exploitation. In the proposed procedure, the individuals are ranked according to their fitness scores from worst to best, thus overcoming the fitness scaling issue. After this, split the whole population into two portions and assigning them probabilities for selection based on their ranks. The χ^2 goodness-of-fit test confirms that there is insignificant difference between the expected and the actual number of offspring. To evaluate the performance of the proposed operator, we conducted a series of simulation study along with some conventional operators. Computational results proved the superior performance of the new selection scheme in comparison with the traditional GA approaches. The significance of such improvement is also validated through two-tailed t test. Hence, the proposed operator might be a good candidate to get optimum or near to optimum results. Moreover, researchers might be more confident to apply it for any problems related to evolutionary algorithms.

References

1. Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, Oxford
2. Osyczka A (2002) Evolutionary algorithms for single and multicriteria design optimization. Springer, Physica-Verlag, Berlin
3. Corriveau G, Guilbault R, Tahan A, Sabourin R (2016) Bayesian network as an adaptive parameter setting approach for genetic algorithms. Complex Intell Syst 2(1):1–22
4. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company, Reading
5. Sivaraj R, Ravichandran T (2011) A review of selection methods in genetic algorithm. Int J Eng Sci Technol 3(5):3792–3797
6. Goldberg DE, Deb K (1991) A comparative analysis of selection schemes used in genetic algorithms. Found Genet Algorithms 1:69–93
7. Pham DT, Castellani M (2010) Adaptive selection routine for evolutionary algorithms. Proc Inst Mech Eng Part I J Syst Control Eng 224(6):623–633
8. Abd-Rahman R, Ramli R, Jamari Z, Ku-Mahamud KR (2016) Evolutionary algorithm with roulette-tournament selection for solving aquaculture diet formulation. Math Probl Eng 2016:1–10
9. Sundar S, Singh A (2010) A swarm intelligence approach to the quadratic minimum spanning tree problem. Inf Sci 180(17):3182–3191
10. Ramli R (2004) An evolutionary algorithm for the nurse scheduling problem with circadian rhythms. Ph.D. thesis, University Sains Malaysia
11. Ramli R, Ibrahim H, Shung LT (2013) Innovative crossover and mutation in a genetic algorithm based approach to a campus bus driver scheduling problem with break consideration and embedded overtime. Appl Math Inf Sci 7(5):1921–1928
12. Yeh HD, Lin CC, Yang BJ (2014) Applying hybrid heuristic approach to identify contaminant source information in transient groundwater flow systems. Math Probl Eng 2014:1–13
13. Asyikin S.N, Razali M (2011) Menu planning model for Malaysian boarding school using self-adaptive hybrid genetic algorithms. Ph.D. thesis, University Utara Malaysia
14. Razali N.M, Geraghty J (2011) Genetic algorithm performance with different selection strategies in solving TSP. In: Proceedings of the world congress on engineering, vol 2. International Association of Engineers Hong Kong, pp 1134–1139
15. Aibinu AM, Salau HB, Rahman NA, Nwohu MN, Akachukwu C (2016) A novel clustering based genetic algorithm for route optimization. Eng Sci Technol Int J 19(4):2022–2034
16. Yan HS, Wan XQ, Xiong FL (2015) Integrated production planning and scheduling for a mixed batch job-shop based on alternant iterative genetic algorithm. J Oper Res Soc 66(8):1250–1258
17. Sharma A, Mehta A (2013) Review paper of various selection methods in genetic algorithm. Int J Adv Res Comput Sci Softw Eng 3(7):1476–1479
18. Mashohor S, Evans J.R, Arslan T (2005) Elitist selection schemes for genetic algorithm based printed circuit board inspection system. In The 2005 IEEE congress on: evolutionary computation, vol 2, pp 974–978
19. Grefenstette JJ (1986) Optimization of control parameters for genetic algorithms. IEEE Trans Syst Man Cybern 16(1):122–128
20. Baker J.E (1985) Adaptive selection methods for genetic algorithms. :In Proceedings of an International Conference on Genetic Algorithms and their applications. Hillsdale, pp 101–111
21. Back T, Fogel DB, Michalewicz Z (2000) Evolutionary computation 1: basic algorithms and operators, vol 1. CRC Press, Boca Raton
22. Blickle T, Thiele L (1995) A comparison of selection schemes used in genetic algorithms. TIK-report
23. Whitley L.D (1989) The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In: ICGA, vol 89. Fairfax, pp 116–123
24. Back T (1996) Evolutionary algorithms in theory and practice. Oxford Press, Oxford
25. Julstrom B.A (1999) It's all the same to me: Revisiting rank-based probabilities and tournaments. In: Proceedings of the congress on evolutionary computation, vol 2. IEEE, pp 1501–1505
26. Eiben A.E, Schut M.C, de-Wilde A.R (2006) Is self-adaptation of selection pressure and population size possible? A case study. In: Parallel problem solving from nature-PPSN IX, pp 900–909
27. Schell T, Wegenkittl S (2001) Looking beyond selection probabilities: adaptation of the χ^2 measure for the performance analysis selection methods in GAs. Evol Comput 9(2):243–256
28. Larranaga P, Kuijpers CM, Murga RH, Inza I, Dizdarevic S (1999) Genetic algorithms for the traveling salesman problem: a review of representations and operators. Artif Intell Rev 13:129–170
29. Ha MH, Bostel N, Langevin A, Rousseau LM (2014) An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size. Comput Oper Res 43:9–19
30. Ho W, Ji P (2009) An integrated scheduling problem of PCB components on sequential pick-and-place machines: mathematical models and heuristic solutions. Expert Syst Appl 36(3):7002–7010
31. Ezziane Z (2006) Applications of artificial intelligence in bioinformatics: a review. Expert Syst Appl 30(1):2–10
32. Applegate DL, Bixby RE, Chvatal V, Cook WJ (2006) The traveling salesman problem: a computational study. Princeton University Press, Princeton

33. Helsgaun K (2000) An effective implementation of the Lin–Kernighan traveling salesman heuristic. Eur J Oper Res 126(1):106–130
34. Huang HX, Li JC, Xiao CL (2015) A proposed iteration optimization approach integrating backpropagation neural network with genetic algorithm. Expert Syst Appl 42(1):146–155
35. Ruiz E, Albareda-Sambola M, Fernandez E, Resende MG (2015) A biased random-key genetic algorithm for the capacitated minimum spanning tree problem. Comput Oper Res 57:95–108
36. Zhang X, Zhang Y, Hu Y, Deng Y, Mahadevan S (2013) An adaptive amoeba algorithm for constrained shortest paths. Expert Syst Appl 40(18):7607–7616
37. Zhang X, Mahadevan S, Deng Y (2015) Physarum-inspired applications in graph-optimization problems. Parallel Process Lett 25(01):1540005
38. Ahmed ZH (2010) Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator. Int J Biometr Bioinform (IJBB) 3(6):96–105
39. Bhattacharyya M, Bandyopadhyay AK (2008) Comparative study of some solution methods for traveling salesman problem using genetic algorithms. Cybern Syst 40(1):1–24
40. Hussain A, Muhammad YS, Sajid MN, Hussain I, Shoukry MA, Gani S (2017) Genetic algorithm for traveling salesman problem with modified cycle crossover operator. Comput Intell Neurosci 2017:1–7
41. Moon C, Kim J, Choi G, Seo Y (2002) An efficient genetic algorithm for the traveling salesman problem with precedence constraints. Eur J Oper Res 140(3):606–617
42. Nagata Y, Soler D (2012) A new genetic algorithm for the asymmetric traveling salesman problem. Expert Syst Appl 39(10):8947–8953
43. Philip A, Taofiki AA, Kehinde O (2011) A genetic algorithm for solving traveling salesman problem. Int J Adv Comput Sci Appl 2(1):26–29
44. Potvin JY (1996) Genetic algorithms for the traveling salesman problem. Ann Oper Res 63(3):337–370
45. Reinelt G (1995) TSPLIB. TSPLIB95 http://www.iwr.uni-heidelberg.de/groups/comopt/software. Accessed 15 July 2018
46. Yuan S, Skinner B, Huang S, Liu D (2013) A new crossover approach for solving the multiple traveling salesmen problem using genetic algorithms. Eur J Oper Res 228(1):72–82