International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Quaternary-Child Crossover for Genetic Algorithm in Real-Time Scheduling Optimization

Furkan Rabee[1] Israa Jazaery[1]* Kamlesh Kumar[2]

[1] *Computer Science Department, Faculty of Computer Science and Mathematics, University of Kufa,Najaf, Iraq*
[2] *Department of Software Engineering, Sindh Madressatul Islam University, Karachi, Pakistan*
* Corresponding author's Email: israas.alhajar@uokufa.edu.iq

Abstract: The optimization method with the genetic algorithm (GA) is well-known in different science fields. The crossover is the primary important operand of the genetic algorithm, so many researchers worked with this operand to get the optimal solutions. This article presents a new crossover operand for the genetic algorithm by supposing a new organization for genes inside the chromosome. The new gene supposed that contains two alleles, and each allele refers to an attitude; thus, each gene has two attitudes. One attitude of the gene supposes to be preferred over another. Therefore, two crossovers were presented, the first crossover applying to intense the preferred one on another as enhancing method of genes to a single individual's chromosome before mating. After the first crossover, the mating process will come by the second crossover, which mating the parents or individuals. The second crossovers' output is four children; eliminate the two lowest fitness, and select the others as new offspring, and they will be part of the next generation. This process is repeated for each population until it reaches the last generation. The new crossover is called the quaternary-child crossover. The proposed is supposed to be applicable for real-time scheduling and is not limited to that. Empirical experiments have tested the proposed work as an optimization method with different mathematical and scheduling equations that had been used in optimization by researchers, and the performance was satisfactory.

Keywords: Real-time scheduling optimization, Genetic algorithm, N-point- crossover, Search optimization.

1. Introduction and background

1.1 Introduction

A genetic algorithm (GA) is a search method to reach the optimal. The biological behavior of chromosomes was the inspiration [1]; the point to make the GA a part of artificial intelligence in recent times. The evolution algorithms follow the evaluation theory of Darwin, called "Survival of the fittest" [1]. Therefore, the strategy for using GA is to get the best offspring (solution). Moreover, this gives the importance of GA. the GA start with two parents and mate between them to get new offspring. This mate accomplishes the crossover. The crossover is essential to get the offspring, then the mutation. Following crossover and mutation operators, the new population replaces the old one. This procedure keeps going till the convergence requirement is satisfied [2].

1.2 GA operands

In typical GA , There are three operands [3], and they are as follows:

1. Selection: The population's chromosomes are chosen for reproduction using this operand. The likelihood of a chromosome being chosen for reproduction increases with its fit.
2. Crossover: This operator randomly selects a locus to produce two children and switches the subsequences between two chromosomes before and after that locus [3].
3. Mutation: This is the process of producing children from a single parent by flipping one or more randomly chosen bits in the parent's chromosomes [4]. Any bit can undergo mutation with a meager chance, like 0.001 [3]. The typical layout Fig. 1 shows the typical GA process

design.

This approach is based on the finding that certain chromosomally encoded features are shared by people and can be inherited by their offspring through crossover [5]. Two children produced have the same genes from one parent or both, with mutation.

1.3 Single–point crossover

This type of crossover was presented by Holland John [6]. This is the most popular and extensively used crossover is the single-point crossover. Along the length of the matched strings, a crossover site is chosen at random, and bits right near the cross-sites are switched. When a proper location is selected, the beneficial traits of the parents might be combined to produce better offspring. When good parents are combined, the offspring will be better if the right location is chosen; otherwise, the string quality will be severely hampered. Suppose one chromosome's head and tail have good genetic material. In that case, none of the offspring will inherit the two favorable traits directly during the single-point crossover.

1.4 N-point crossover

The n-point crossover was first implemented by De Jong [7]. Although there are multiple crossover points, the rule is the same as the one we used for single-point crossover. The relevance of the crossing sites in a two-point crossover is two. The interruptions of building blocks caused by continually adding crossover sites can cause the performance of genetic algorithms to suffer.

1.5 Operators definition

Many operators refer to the main parts of building a genetic algorithm form; a gene is a string of (bit or real number) of arbitrary length. A sequence of genes is called a chromosome. The smallest unit in chromosomes is called an allele, which can be represented by a symbol or bit. A genotype is a piece of information stored in a chromosome, but a phenotype provides an exterior description of the individual [4, 8]. Fig. 2 shows the primary operand of GA [4, 8].

In this proposed work, a new organization for the chromosome's genes was supposed that every two alleles of the chromosome should be grouped as one gene with two attitudes. And suppose that one of these attitudes could be preferred and the other does not prefer. A new crossover has been presented to enhance one of the gene's attitudes on the cost of the author.

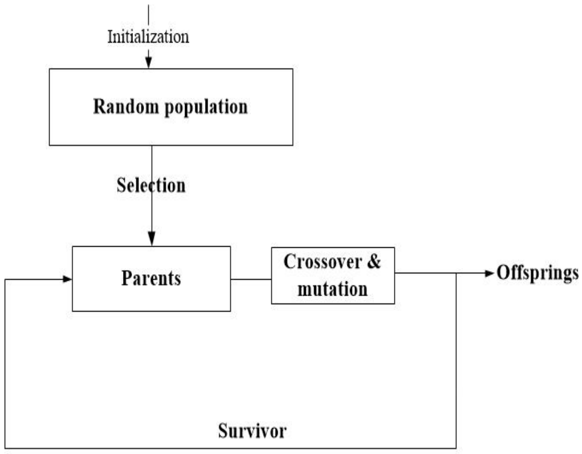


Figure. 1 Typical genetic algorithm design

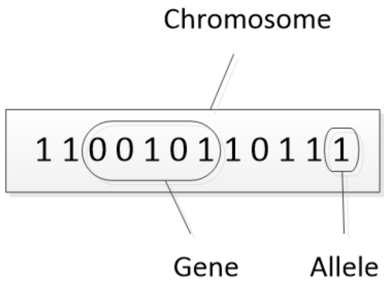


Figure. 2 The GA operators

2. Related works

2.1 Original theories

Gilbert Syswerda [9] first presented the uniform crossover for GA, even the one point, the second point was presented, but the uniform crossover showed its outperform in optimization, and till now, this crossover type is applicable in many different science fields.

William M. Spears [7] presented an adaptive algorithm to decide when a particular crossover(one point, second point, or uniform) will be optimal for any problem. However, it still works with standard crossover. this article shows that optimization for all systems, but still, there is a drawback for improving specific parameters.

Umbarkar, Anant J Sheth, and Pranali D [10] presented an excellent review showing more than thirty-five types of crossover presented till 2015, the researches used in different optimization fields. But no one could optimize the preferred parameters and limit the un-prefer parameters in the system.

Chiroma Haruna, Abdulkareem Sameem, Abubakar Adamu, and Herawan Tutut [4] presented a review showing GA's importance in optimization for Machine Learning and Deep learning. Even though, they don't find any change in chromosome

 DOI: 10.22266/ijies2023.0430.09

Received: November 26, 2022. Revised: December 26, 2022.

organization in the revised research.

Abid Hussain, Yousaf Shad Muhammad, and Muhammad Nauman Sajid [2] presented a schema that extended to the two-point crossover, where the proposed approaches provide a comparative convergence rate. Even though this article presented a modification for the crossover, they don’t change the architecture of the chromosome itself.

Luca Manzoni, Luca Mariot, and Eva Tuba [5] presented balanced crossover operators who ensure that the offspring has the same balanced characteristics as the parents. This article doesn’t add any change to the number of offspring and the chromosome structure, even if it equalized the chance of the offspring's characteristics, and we think it benefits less in many systems optimization.

[4] presented a modified optimization method depending on AI. Presented guidance for both beginner and experienced researchers designing evolutionary neural networks, assisting them in selecting appropriate genetic algorithm operator values for use in applications in a certain issue domain.

2.2 Implementation of GA

There are too many articles that used GA, but here we mentioned below the essential and recent years to show the importance of GA in many fields, they are as follows:

Nicolas Kirchner-Bossi, and Fernando Porté-Agel [11] presented a model for reducing the power losses associated with turbine wakes in wind farms; this model did through a genetic algorithm.

Félix Patrón, Roberto Salvador Botez, Ruxandra Mihaela [12] presented a new flight trajectory calculation using GA. the method analyzed lateral and vertical navigation to obtain the optimal cruise trajectory in terms of fuel consumption, where the optimization via GA saved fuel up to 5.6%.

Dana Bani-Hani, Naseem Khan, Fatimah Alsultan, Shreya Karanjkar, and Nagen Nagarur [13] used deep learning’s convolutional neural network to classify four types of leucocytes. A genetic algorithm was used to optimize the CNN's hyperparameters (GA); this article showed that CNN is not efficient in getting optimal.

Fatemeh Ahmadi Zeidabadi and Mohammad Dehghani [14] presented a new optimization method called puzzle optimization algorithm(POA), which can be used in different optimization problems. The advantage of this method is that there are no control parameters, thus not requiring parameter sitting.

Khandelwal Anju and Kumar Avanish [15] presented a technique based on fuzzy triangular numbers that have been applied to the recruitment process of the individual to the employee. Moreover, the genetic algorithm used for optimization, where the author presented a solving for the selection process to the individual through GA and fuzzy ranking.

Safira Begum, Sunita S Padmannavar [16], their research was with Educational Data Mining (EDM), where they presented a student predictive model using ensemble classifiers and implemented their search with pre-pressing, then classified with data mining technique. Then they used GA to find the optimal solution, search for problems, and increase the probability of solving them.

Haldulakar Rupali and Agrawal Jitendra [17] presented a novel method in one of the vital part in Data mining which is strong rule generation, where the author used differently to generate rules. However, the genetic algorithm was the best optimization way to generate rules.

Shatha Abdulhadi Muthana, and Ku Ruhana Ku-Mahamud [18], they compared multi-objective optimization methods to obtain the generator maintenance scheduling solution. One of the optimization methods was non-dominated sorting genetic algorithm; this confirms that using GA and optimization is very important in different fields of science.

2.3 The importance of optimization

The proposed work in this article supposes to be for many applications in the field of optimization, such as computer networks, routing algorithms, wireless sensor networks, operating system scheduling, real-time operating system scheduling …etc., and not limited to that. Different applications required optimization; we displayed some samples from recent research as follows:

One important field for optimization is cloud computing, where [19] used the optimization technique to reduce the utilization of the energy of the data center in the way to accomplish the quality of service.

In the field of WSN, the energy consumption of nodes should be restricted. Therefore, [20-22] presented an energy-efficient deployment method for sensor nodes by clustering the nodes and applying the optimization technique for optimal reduction of the nodes' battery. The optimization for nodes led to the optimization routing protocol.

Paolo Cinat, Giorgio Gnecco, and Marco Paggi [23], Depending on GA in Mechanical engineering, used in multi-scale surface roughness, presented three genetic algorithms to superimpose and merge the mathematical description of chromosomes to determine the best roughness features.

2.4 Problem statement

From the previous review, we discovered a drew back in the genetic algorithm, where it cannot improve the preferred parameters in the system on the un-preferred parameters. This case could be found in the real-time scheduling system and routing algorithms as shown in this article. Therefore; this article presents new modifications for the genetic algorithm core.

2.5 Aim and objectives

The first modification in the GA core is the mutation process, which could come before crossover or after crossover or can apply it in both times, depending on the application. Where the mutation in the previous related works comes one time after the crossover. The second modification is the output from the mating (by the crossover) is four children, whereas the normal crossover output for the genetic algorithm is two children. Overall, the proposed work presented a new optimization method.

3. The application and methods

This section consists of the following:

3.1 Application

The field of interest and applicable for this article is real-time scheduling (RTS); the critical point in RTS is the system should be accomplished its task within the deadline; elsewhere, the system will fail. So, optimization is essential to prevent the system from missing the deadline.

3.1.1. Allocation algorithm

The rate monotonic best fit (RMBF) has been used in this article, which defines the tasks allocation to the processors with the smallest available utilization, on which it can be feasibly scheduled[24]. RMBF orders task increasingly according to their periods before the allocation process. Tasks can be feasibly scheduled according to IP (increasing period) schedulability condition [24].

The schedulability condition IP requires that the tasks' periods are ordered increasingly; thus, the timing constraints of the task set must be known a priori[25].

The condition IP is, let $T = \{T_1, T_2, \dots, T_n\}$ Be a set of tasks with $T_1 \le T_2 \le \cdots \le T_n$, And E is execution time.

Let

$$u = \sum_{i=1}^{n-1} \frac{E_i}{T_i} \le (n -)\left(2^{\frac{1}{(n-1)}} - 1\right) \quad (1)$$

If the following condition is met,

$$\frac{E_n}{T_n} \le 2\left(1 + \frac{u}{n-1}\right)^{-(n-1)} - 1 \quad (2)$$

Then the set of tasks will have a feasible schedule under real-time scheduling.

3.2 System definitions

Suppose that a population P with N chromosomes where each chromosome with z genies with two alleles as minimum length or gene: suppose that the alleles of genes represented two properties X, and Y. these properties refer to two attitudes, supposed to be one is +ve (or admire) referred by X, and the –ve (or dis-admire) referred by Y. the attitudes represented by the following equations:

$$X_G^n = \{X_1^n, X_2^n, \dots\dots, X_i^n\} \quad (3)$$

$$Y_G^n = \{Y_1^n, Y_2^n, \dots\dots, Y_i^n\} \quad (4)$$

Moreover, combine the two properties as follows:

$$H_n = \{X_1^n Y_1^n, X_2^n Y_2^n, \dots\dots, X_i^n Y_i^n\} \quad (5)$$

Where H refers to the specific Chromosome, G refers to the related gene.

The structure of the proposed chromosome is presented in Fig. 3.

These properties for the H chromosome can be affected by mutation (μ) [1][26, 27]. The mutation (μ) come with two values, one as positive (γ) mutation and other as negative (δ) mutations, which mean the +ve mutation effect by +ve attitude(X) to specific gene, and the –ve mutation which effect by –ve attitude (Y) of to the specific gene. Usually, μ come in two bits to be compatible with the gene. Suppose that the mutation (μ) comes randomly (positive or negative), as follows:

$$\mu|_{\gamma,\delta} = \begin{cases} \gamma_1, \gamma_2 \dots\dots, \gamma_n \\ \delta_1, \delta_2 \dots\dots, \delta_n \end{cases} \quad (6)$$

[1] This type of mutation similar to biological mutation caused by the outside effect such as medicine or drugs or others. Simulate with that, we suppose the mutation could happen with parents before mate.

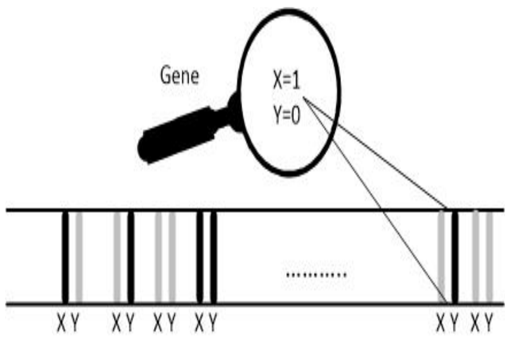


Figure. 3 The proposed chromosome

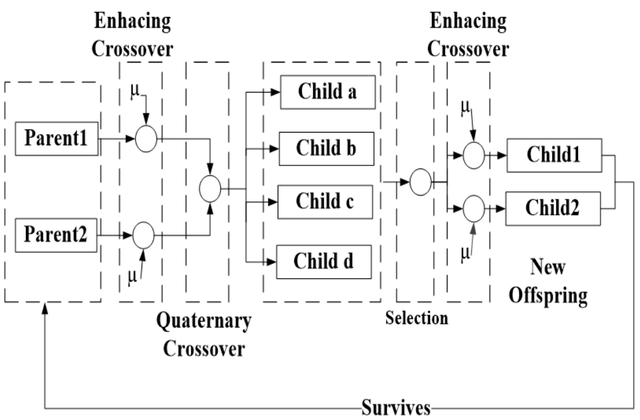


Figure. 4 (Quaternary-children)'s algorithm

Where;

$$\gamma|\delta \ \forall_{\mu} = \begin{cases} 10 + ve \ \ change \\ 01 \ -ve \ change \\ elsewhere \ \ no \ change \end{cases} \quad (7)$$

In Eq. (7) the value of μ, if come 00, or 11, would not affect the gene.

3.3 (Quaternary-children)'s algorithm (QA)

This section describes the proposed system, which consists of many parts. The first part refers to the parents' selection, which depends on the roulette wheel to select the best parent. The second part, enhancing crossover (EC), shows the mutation that exposed the selective parents; this mutation was accomplished by proposed crossover equations called enhancing crossover (EC). It could enhance the parents' attitudes before mating. The EC comes one time for the system for parents or children. A sequenced crossover between the parents represents the third part. Due to the two-bit genes, the offspring becomes four children. The four-part for the system as offspring best selecting, to choose the best two children depending on their fitness, to get the new offspring. If the result is not acceptable, the system will start again. This system is called (quaternary-children)'s Algorithm, as shown in Fig. 4.

3.4 Enhancing crossover (EC)

This type of crossover comes one time in this system, and if comes with a parent, don't come with children; if it doesn't come with a parent, it can come with children. The following equation represents this type of crossover:

$$H_n^T = \begin{bmatrix} X_1^g Y_1^g \mid \gamma_1 \delta_1 \\ X_2^g Y_2^g \mid \gamma_2 \delta_2 \\ \dots \\ \dots \\ X_n^g Y_n^g \mid \gamma_n \delta_n \end{bmatrix} \quad (8)$$

Where g refers to the gene related to the specific H chromosome, and T refers to transport for this matrix to get the vector representing the chromosome. After implementing Eq. (8), the new genes (or enhanced chromosomes) should be getting for the same single parent. The mutation μ carries a +ve or –ve effect to improve the +ve (or admire) attitude and reduce the –ve (or dis-admire) attitude for the chromosome, thus improving the system's preferred parameters. The proposed mutation has little effect on the parent because depending on the Boolean operation.

3.5 Sequenced crossover (SC)

Also, define the quaternary crossover, this crossover for mating the parents, to produce four children as flows following equations:

$$N_{1,n} = \left[H_{2,1} + 1, H_{2,2} + 1, \dots, H_{2,n} + 1\right] \quad (9)$$

$$N_{2,n} = \left[N_{1,1} + 1, N_{1,2} + 1, \dots, N_{1,n} + 1\right] \quad (10)$$

$$N_{3,n} = \left[N_{2,1} + 1, N_{2,2} + 1, \dots, N_{2,n} + 1\right] \quad (11)$$

$$N_{4,n} = \left[N_{3,1} + 1, N_{3,2} + 1, \dots, N_{3,n} + 1\right] \quad (12)$$

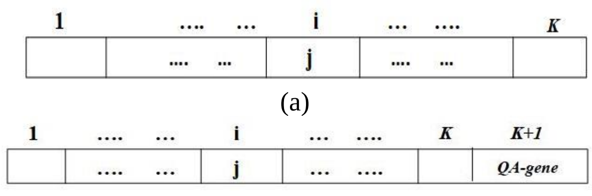


Figure. 5 Activity list presentation: (a) Activity list presentation and (b) Activity list presentation with QA-scheduling

Received: November 26, 2022. Revised: December 26, 2022.

Where N refers to the new child, this system produces four children. Also, in the above equations, as shown, the next child is generated from the one before him by one sequence. The best two children would be chosen depending on their finesses. Section 5.2 show an example for one scenario to explain more detail about the proposed algorithm.

4. QA in real-time scheduling

4.1 Scheduling definitions

The proposed work could work with different fields of science. However, it suggested applying real-time scheduling to choose the best solution for activity list presentation[28]. A list of the activities with precedence that is possible encodes the solution. After all of its predecessors, any action may appear in any position on the list.

Activity list representation: According to GA genotype sequence, each individual in the population is represented by an array with as many positions as activities in the task. Fig. 5 shows the activity list for a task with K activities. Activity j will be the ith activity chosen to be scheduled. It will be scheduled in its earliest feasible start time. When activity j is chosen to be scheduled, all its predecessors, which will appear in some position 1, . . . , i − 1, will have already been scheduled. In this way, the related schedule will always be a feasible schedule.

Notice that when applying this procedure, one and only one schedule (phenotype) can be deduced from a given sequence (genotype), but different sequences could transform into the same schedule. When applying the serial method to transform the representation into a schedule, the search space is formed by the set of active schedules, which always contain an optimal solution[28]. Fig. 6 show a job example.

4.2 Task scheduling example:

Fig. 6 shows a task example with eight jobs, one resource, and six availability per period.

Solution: there are two solution and not limited to that A, and B, and suppose that A, and B is an individual related to the GA.

A

B

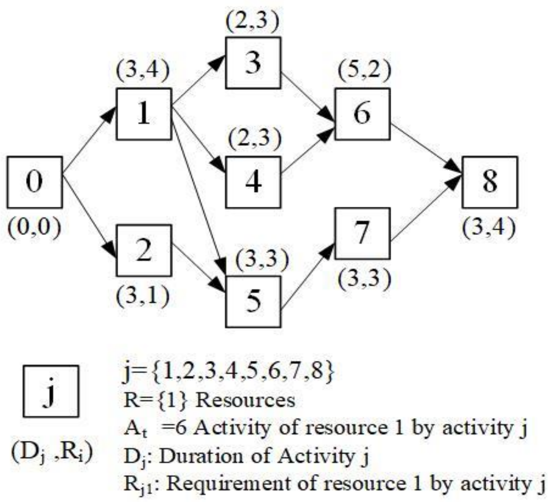
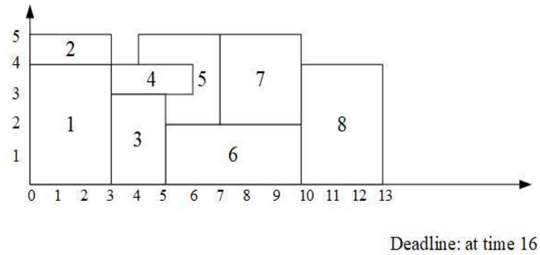
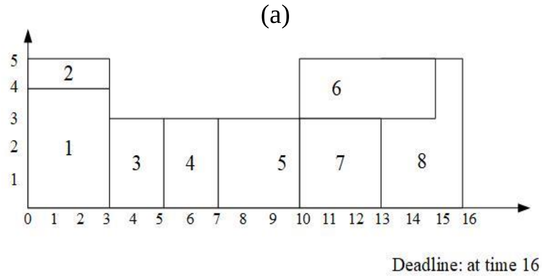


Figure. 6 Task scheduling jobs example



(a)



(b)

Figure. 7 Task scheduling: (a) Schedule solution A with task period =13 and (b) Schedule solution B with task period =16

Fig .7 shows the Gantt chart for the task scheduling for the A, and B solutions. Where A, and B is the optimal solution for this scheduling to meet the scheduling deadline. The QA proposed to optimize such a problem.

From Fig. 6, job (0) represents to source dummy activity job.

From the above example, this system can be implemented with proposed QA, where suppose that each solution (A, and B) refers to an individual(offspring) of the population from pre-selected parents. Each element in (A, and B) has two properties (Dj, Ri); thus, one can choose the optimal solution with deadline constraints with Eq. (1) and Eq. (2).

5. Experimental work

5.1 Optimization implementation procedures

This part shows the practical implementation of the proposed system. The proposed algorithms are used as an optimization method. Our procedures go with the following steps:

1. Initialization for population chromosomes starts by generating random individuals as initialization, each individual's chromosome with ten alleles and five genes, as proposed in section 4.
2. Calculating the cost function for each parent depends on which equation needs to be optimized. The cost function means using the individual as a parameter in the equation needed to optimize. Four equations depended on testing the proposed work. For each equation, there are four isolated pupations have been generated and tested. The equations are as follows:

2.1 General optimization test

Here displayed two equations that have been used for general mathematical optimization, they are:

2.1.1 Uneven Decreasing Maxima Function[2, 29]:

this is one of the multimodal optimization problems.

$$F = e^{\left(-2\log(2)\left(\frac{\emptyset-0.08}{0.854}\right)^2\right)} sin^6\left(5\pi\left(\emptyset^{\frac{3}{4}} - 0.05\right)\right) \quad (13)$$

Where, $\emptyset \in [0,1]$. This optimization for this equation minimizes F

2.1.2 Himmelblau function:

As mathematical optimization, Himmelblau's function, which is used by [2, 30] , defines:

$$F = 200 - (\emptyset_1^2 + \emptyset_2 - 11)^2 - (\emptyset_1 + \emptyset_2^2 - 7)^2 \quad (14)$$

Where, $\emptyset_i \in [-6,6]$. This optimization for this equation minimizes F.

2.1.3 Communication and scheduling optimization

Two equations depend on this article. These equations are used to optimize the route and scheduling operation for real-time resources [28]

$$F = \sum_{i=1}^{n} x_i y_i \quad (15)$$

$$F = \sum_{i=1}^{n} (x_i y_i)^2 \quad (16)$$

The optimization for the above equations is to minimize the value of F.

3. Choose ten individuals to represent the first generation, then calculate the fitness of this generation or generation fitness (G_T) as follows:

$$G_T = \sum_{i=1}^{n} C_i^p \quad (17)$$

Where; n: number of individuals in that generation, C_i^p , Referring to the individual p cost function, the fitness is calculated by applying Eq. (13) to Eq. (17). Then calculating the fitness of each individual as follows:

$$f_i^p = \frac{C_i^p}{G_T} \quad (18)$$

4. Calculate the probability for each individual as follows:

$$Prob_i = f_i^p * n \quad (19)$$

Then rearranging the individual depends on the probability of choosing the best individual to be a good selected parent in this generation.

5. After arranging the individual in decreasing order, eliminate the last two individuals (the two least probability).
6. After eliminating two individuals, it now has the best parents ready to mate; thus, it makes the crossover according to the proposed QA algorithm to get new offspring as a new generation.
7. Repeat the steps from (2) to (6) again till they reach only two individuals in the offspring.
8. Repeat steps (1) to (7) for one hundred iterations to choose optimal values.
9. The steps from (1) to (8) are implemented with four Eq. (13) to Eq. (16).

5.2 Scenario for QA

Suppose there are two higher probability parents with their chromosomes as follows;

$H_{1,3} = \{00\,,10\,,00\}$, $H_{2,3} = \{01, 01, 01\}$, and μ_1={01, 00, 11}, μ_2={11, 10, 01}. Apply the QA; suppose the enhanced crossover (EC) disposed of the parents. Then show the first offspring, suppose that the cost function for chromosome can get it by the summations of chromosome's alleles as the equation: $C = \sum_{i=1}^{n} l_i$.

Solution

To apply the EC, should apply Eq. (8) with conditions in Eq. (7), and the new parents should do as follows;

$H_{1,3}^{new} = \{01\,,10\,,00\}$, $H_{2,3}^{new} = \{01, 11, 01\}$.

and to calculate the fitness for each parent applying Eq. (17) and Eq. (18) equations.

So, the cost function for $H_{1,3}$ = 2, fitness = 0.333 and the $H_{2,3}$ = 4, and fitness= 0.666. Now apply the Eq. (9-12) as follows:

$N_{1,3} = \{10, 00, 10\}$ with cost function = 2 , fitness=0.166.

$N_{2,3} = \{11, 01, 11\}$ with cost function = 5 , fitness=0.416.

$N_{3,3} = \{00, 10, 00\}$ with cost function = 1 , fitness=0.083.

$N_{4,3} = \{01, 11, 01\}$ with cost function = 4 , fitness=0.333.

From the above result, the best children are $N_{2,3}$ and $N_{4,3}$, so they are the new offspring become.

5.3 Recursive process

The experimental work is done by many recursive processes, which means that the output from any generation will be input for the next generation. However, it depends on which crossover used to be the reference to be the input for the next generation. The recursive process for generations depends on their finesses. Fig. 8 shows the flow work of the recursive processes.

In this article, the experimental work was tested for five generations. The symbols in Fig. 8 mean the initialization refers to the parent selection process for the parents; after selection. The blue ball refers to the initialization fitness values which tested all types of crossover methods. The output will give two balls red and green, where the red refers to the output after applying N-Crossover methods, and the green refers to the output after applying the QA crossover.

Fig. 8-A shows the QA fitness as a recursive fitness input for the next generation. Fig. 8-B shows the N-Crossover fitness as a recursive fitness input for the next generation. The benefits of using different recursive methods, are to show the influence of the proposed method even applying the shortcoming output from N-crossover. The proposed method shows its outperforming in two methods. Where N was used with different values. The steps in section 5.1 were implemented for N-crossover and QA crossover with each generation.

5.4 Experimental calculations

5.4.1. Genetic algorithm parameters

Many experiments have been implemented. All the algorithm has been programmed from scratch via Matlab 2016a. Some of the function parameters bounders we could not commit with it because they may conflict with a binary crossover on which we depend and propose according. Table 1 shows that the GA parameter has been dependent on this proposed work. This article depends on n-sections-crossover because this type, the closest to the proposed work, and compared with other crossovers, got meaning less even the proposed work outperformed. The empirical results perform in two

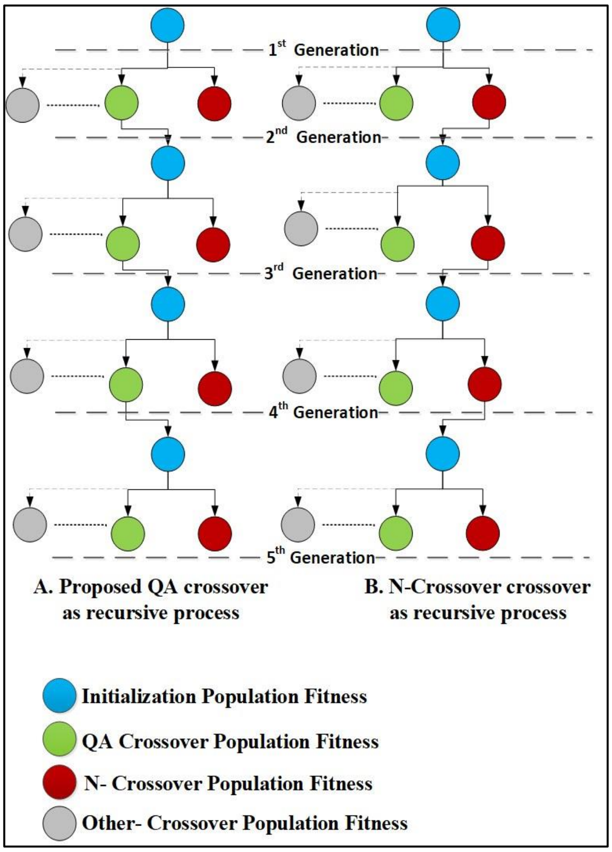


Figure. 8 The recursive processes for generations' fitness's

Table 1. Genetic algorithm parameters

Parameters	Value per Population	Factor
Population Size	10	x 12
Scaling Function for selection probability	Uniform distribution	………
Selection Operator	Roulette Wheel	………
Crossover Probability	80%	x 12
Mutation Operator	xor	………
Mutation Probability	10%	………

Received: November 26, 2022. Revised: December 26, 2022.

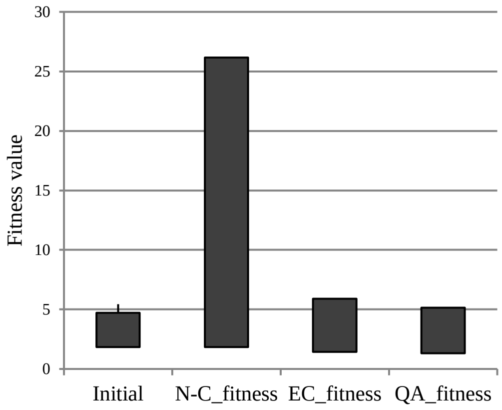


Figure. 9 Fitness comparison with Eq. (13) and n-crossover recursive

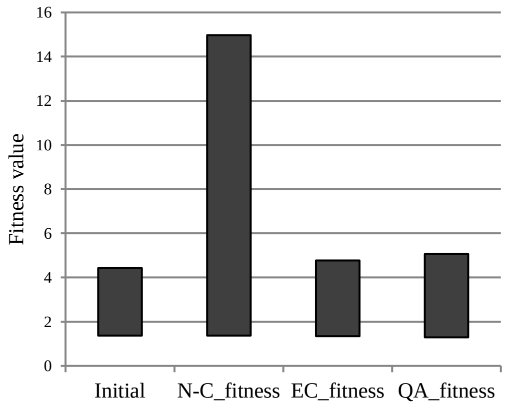


Figure. 10 Fitness comparison with Eq. (13) and EC recursive

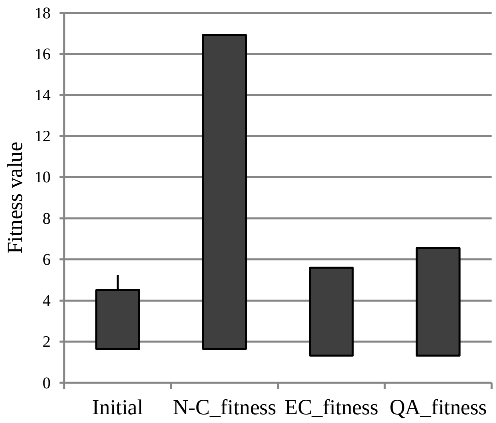


Figure. 11 Fitness comparison with Eq. (13) and QA recursive

ways, generations fitnesses calculation comparison and equilibrium generations fitnesses over one hundred iterations.

5.4.2. Generations fitness's comparison

This section presented the experiments that were done to show the performance of the proposed system, the fitnesses comparison for the proposed system with implemented four Eq. (14) to Eq. (17), on the n-section crossover, enhanced crossover (EC), and sequenced crossover (SC). With three recursive processes, n-section crossover recursive, enhanced crossover recursive, and sequenced crossover recursive. Fig. 9 to Fig. 20 shows generations fitness's comparison represented by the japanese candlestick charts, where the white candlesticks refer to increased quantities, and dark candlesticks refer to decreased quantities.

Fig. 9 to Fig. 11 show that EC and QA in decreasing values, and this is the goal of Eq. (13), where the optimization of this equation to get the minimum values and the proposed accomplished the required mission better than others.

Fig. 12 to Fig. 14 show that QA is increasing values because of the optimization for Eq. (14) for maximization. In this work, we don't commit to the boundaries for this equation because we used the binary crossover and got some conflict with the negative values of the boundary values of this equation. Even the proposed work accomplishes the mission of maximization better than the others.

Fig. 15 to Fig. 20 show that the proposed work is minimization optimization, and this is the purpose of Eq. (15) and Eq. (16).

One hundred iterations have been done to choose wherever the system will reach equilibrium optimization for generations' fitnesses. The test was done in three recursive ways, as mentioned before. Fig. 21 to Fig. 32 show the results of these iterations and that the proposed algorithms have done their jobs as required.

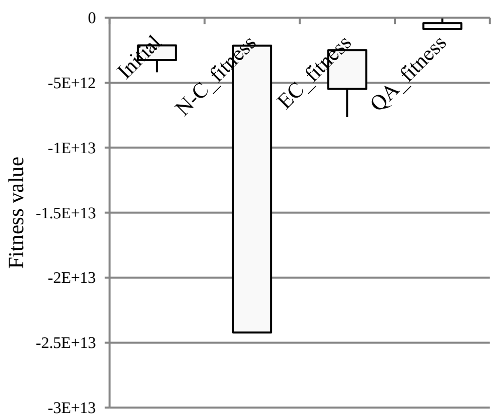


Figure. 12 Fitness comparison with Eq. (14) and n-crossover recursive

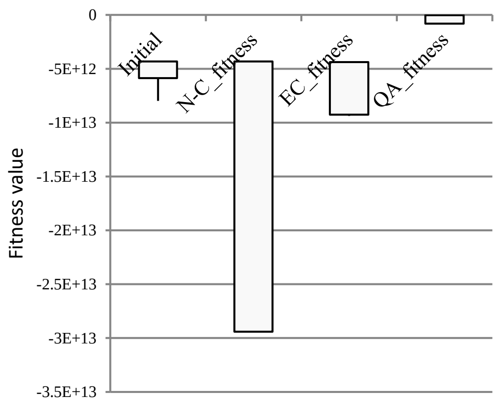


Figure. 13 Fitness comparison with Eq. (14) and EC recursive

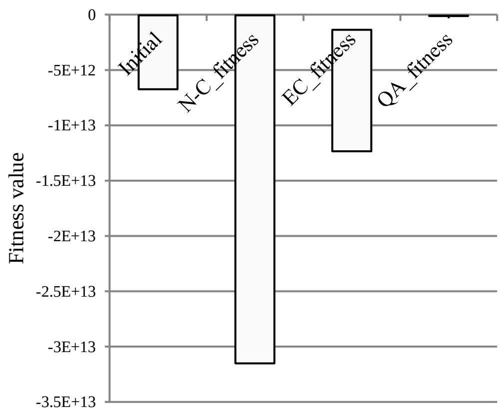


Figure. 14 Fitness comparison with Eq. (14) and QA recursive

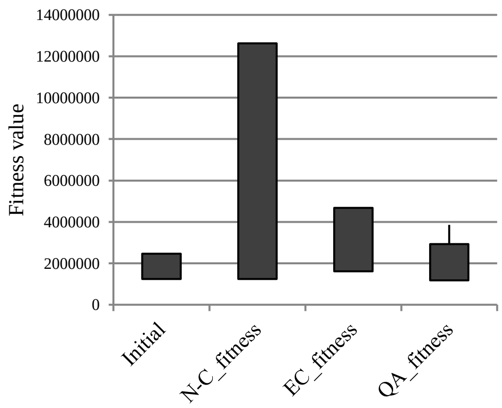


Figure. 15 Fitness comparison with Eq. (15) and n-crossover recursive

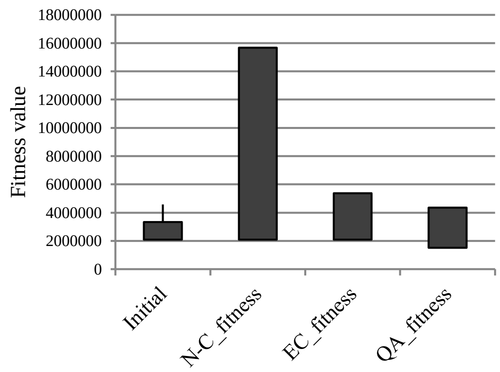


Figure. 16 Fitness comparison with Eq. (15) and EC recursive

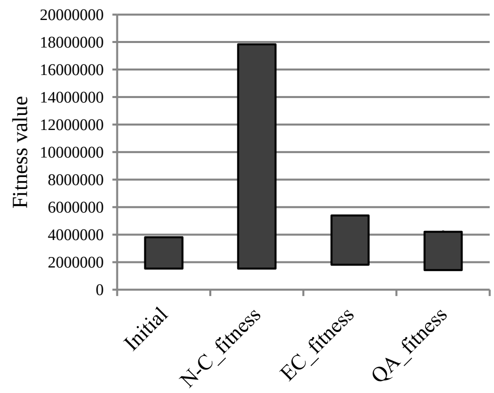


Figure. 17 Fitness comparison with Eq. (15) and QA recursive

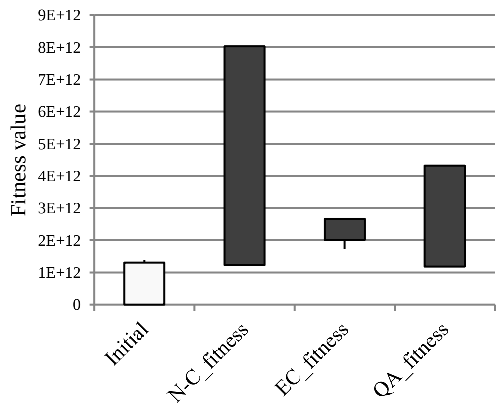


Figure. 18 Fitness comparison with Eq. (16) and n-crossover recursive

Received: November 26, 2022. Revised: December 26, 2022.

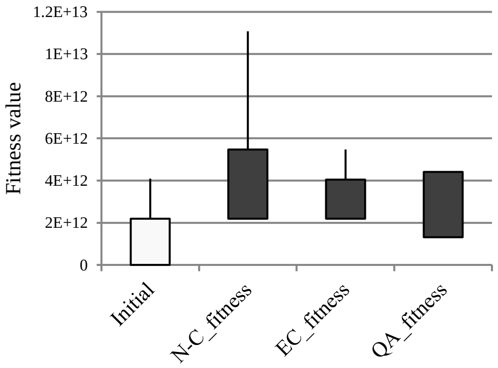


Figure. 19 Fitness comparison with Eq. (16) and EC recursive

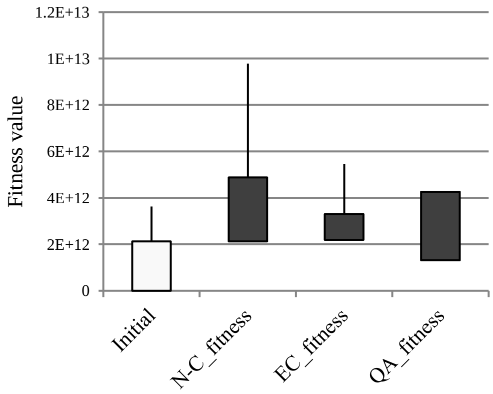


Figure. 20 Fitness comparison with Eq. (16) and QA recursive

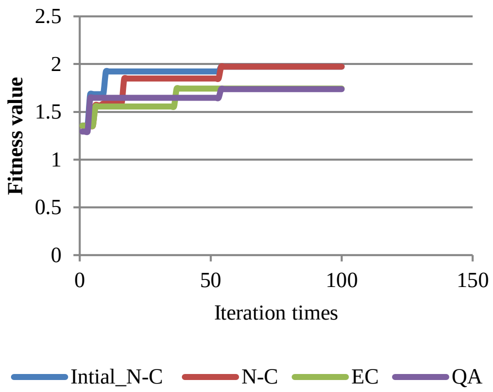


Figure. 21 Optimization equilibrium with Eq. (13) and n-crossover recursive

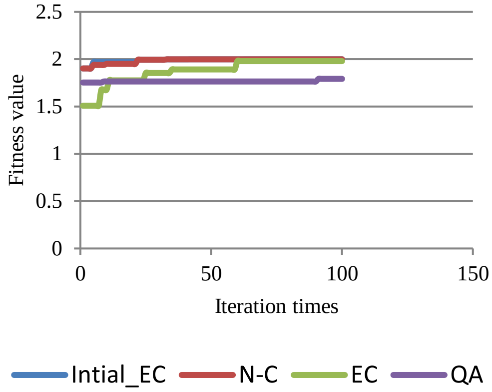


Figure. 22 Optimization equilibrium with Eq. (13) and EC recursive

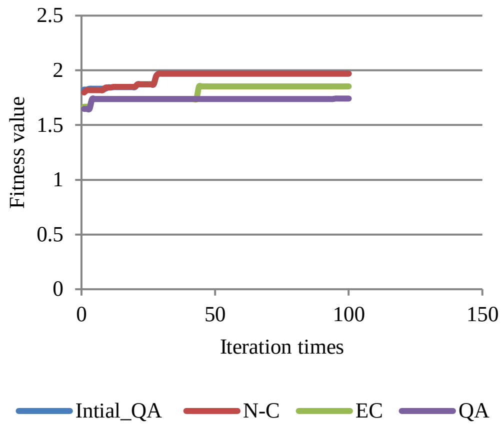


Figure. 23 Optimization equilibrium with Eq. (13) and QA recursive

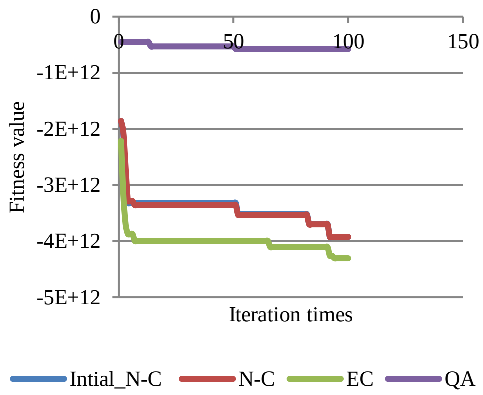


Figure. 24 Optimization equilibrium with Eq. (14) and n-crossover recursive

 DOI: 10.22266/ijies2023.0430.09

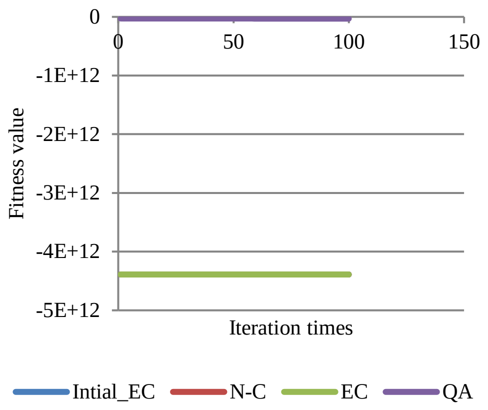


Figure. 25 Optimization equilibrium with Eq. (14) and EC recursive

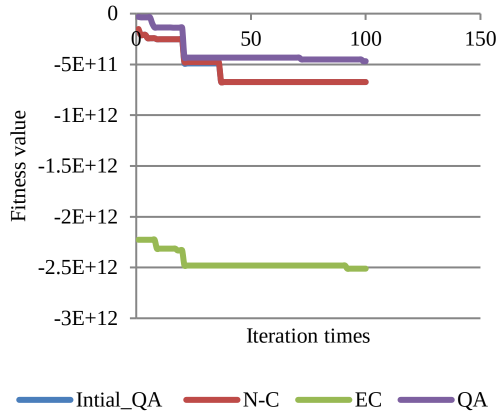


Figure. 26 Optimization equilibrium with Eq. (14) and QA recursive

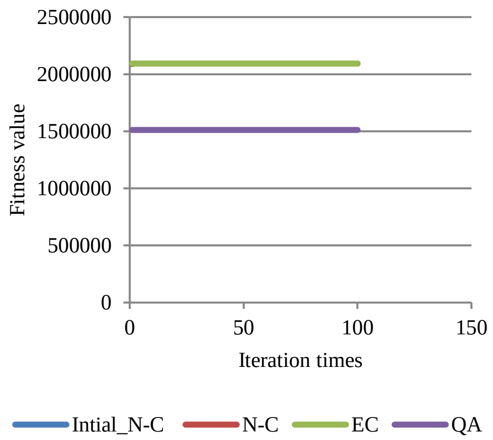


Figure. 27 Optimization equilibrium with Eq. (15) and n-crossover recursive

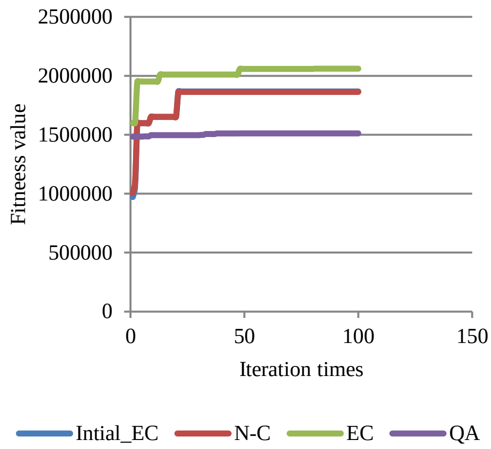


Figure. 28 Optimization equilibrium with Eq. (15) and EC recursive

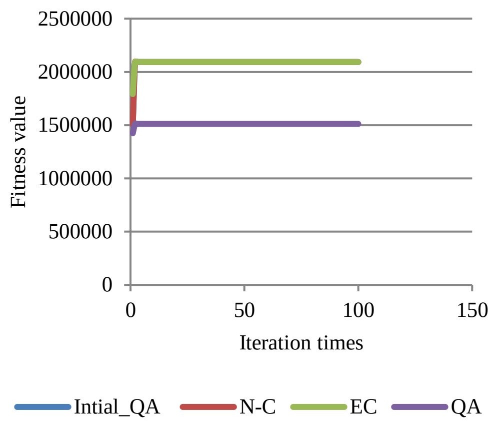


Figure. 29 Optimization equilibrium with Eq. (15) and QA recursive

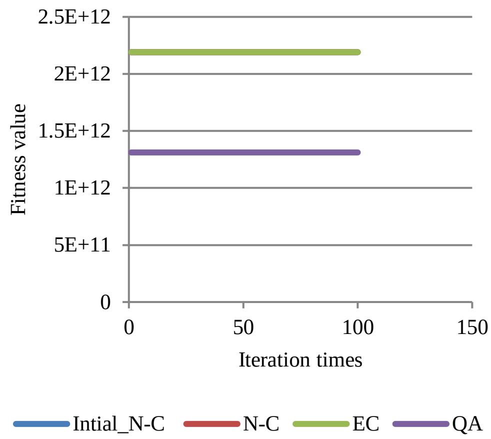


Figure. 30 Optimization equilibrium with Eq. (16) and n-crossover recursive

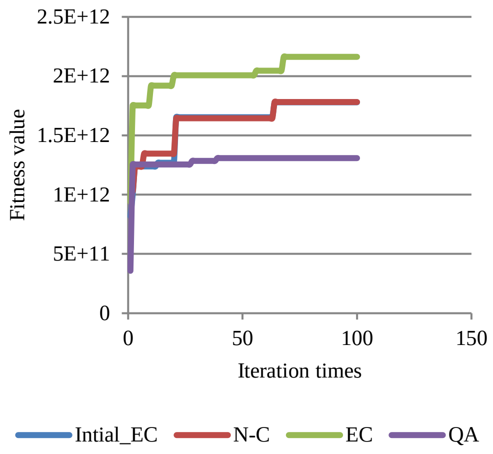


Figure. 31 Optimization equilibrium with Eq. (16) and EC recursive

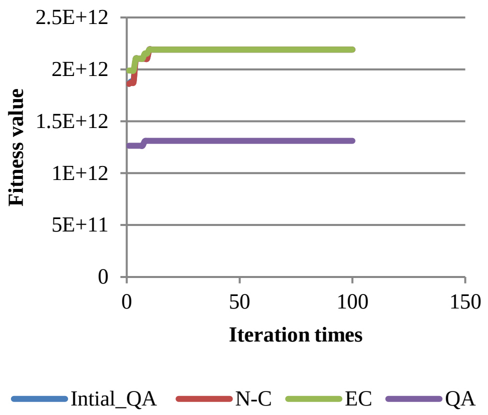


Figure. 32 Optimization equilibrium with Eq. (16) and QA recursive

6. Conclusion

From the proposed theoretical and experimental work, we conclude that:

1. The proposed system and algorithms are inspired by human genetics, so this could be near to the natural characteristic of human life. This leads to the optimization way that would be more accurate for scientific applications.
2. Depending on two attitudes in the proposed optimization shows a new way to improve the preferred attitude for some parameters in many fields, such as control systems, computer networks, routing protocols, real-time scheduling, operating system, and others.
3. Extracting four children from the crossover-inspired Mendel theory, skin color as an example, and this gives a wide area to select better offspring. This is a new selection method after the crossover.
4. The quaternary-child crossover shows the ability to get more than two children from two parents.
5. The recursive methods used in experimental work show that depending on the same decent related to the same crossover method gives more optimization. For example, applying the QA crossover from the first generation to the final generation, the final QA offspring gave a better optimization result.
6. From the results, with Fig. 9 to Fig. 11, and with Fig. 15 to Fig.20, the proposed work shows its effectiveness in this field of optimization, and this optimization is very suitable with real-time scheduling and routing algorithms in a computer network.
7. The result from comparing the proposed work with n-point-crossover concludes our proposal, such as optimizing the method for n-point crossover. So the QA could be a better alternative crossover for n-point crossover.

Conflicts of interest

"The authors declare no conflict of interest".

Scientific contribution

The proposed algorithms present a new optimization method, by proposing a new core modification for the crossover and mutation operands to the genetic algorithm. Thus, this would help the researchers to use this system in many fields for optimization, especially real-time scheduling, and routing algorithms.

Author contribution

The paper conceptualization, Furkan Rabee, Methodology, Furkan Rabee; Software, Furkan Rabee, and Israa; formal analysis, Furkan Rabee, and Kumar; investigation, Furkan Rabee, and Israa Jazaery; resources, Furkan Rabee, data curation, Furkan Rabee, writing original draft preparation, Furkan Rabee; writing-review and editing; Kamlesh Kumar; visualization, Israa Jazaery, Kamlesh Kumar; Supervision, Furkan Rabee; project administration Furkan Rabee and Israa Jazaery.

References

[1] P. Kora and P. Yadlapalli, "Crossover operators in genetic algorithms: A review", *International Journal of Computer Applications*, Vol. 162, 2017.

[2] Y. S. M. A. Hussain and M. N. Sajid, "An Efficient Genetic Algorithm for Numerical Function Optimization with Two New

Crossover Operators", *International Journal of Mathematical Sciences and Computing*, Vol. 4, pp. 1-17, 2018.

[3] M. Mitchell, An introduction to genetic algorithms: MIT press, 1998.

[4] H. Chiroma, S. Abdulkareem, A. Abubakar, and T. Herawan, "Neural networks optimization through genetic algorithm searches: a review", *Appl. Math. Inf. Sci*, Vol. 11, pp. 1543-1564, 2017.

[5] L. Manzoni, L. Mariot, and E. Tuba, "Balanced crossover operators in genetic algorithms", *Swarm and Evolutionary Computation*, Vol. 54, p. 100646, 2020.

[6] H. H. John, "Adaptation in natural and artificial systems", *Ann Arbor: University of Michigan Press*, 1975.

[7] W. M. Spears, "Adapting crossover in a genetic algorithm, in: J.R. McDonnell,R.G. Reynolds, D.B. Fogel (Eds.)", In: *Proc. of the Fourth Annual Conference on Evolutionary Programming*, pp. 367–384,1995.

[8] S. Sivanandam and S. Deepa, "Genetic algorithms", *Introduction to Genetic Algorithms*, ed: Springer, 2008, pp. 15-37.

[9] G. Syswerda, "Uniform Crossover in Genetic Algorithms", In: *Proc. of the 3rd International Conf. on Genetic Algorithms*, 1989.

[10] A. J. Umbarkar and P. D. Sheth, "Crossover operators in genetic algorithms: a review", *ICTACT Journal on Soft Computing*, Vol. 6, 2015.

[11] N. K. Bossi and F. P. Agel, "Realistic wind farm layout optimization through genetic algorithms using a Gaussian wake model", *Energies*, Vol. 11, p. 3268, 2018.

[12] R. S. F. Patrón and R. M. Botez, "Flight trajectory optimization through genetic algorithms coupling vertical and lateral profiles", in *ASME International Mechanical Engineering Congress and Exposition*, p. V001T01A048, 2014.

[13] D. B. Hani, N. Khan, F. Alsultan, S. Karanjkar, and N. Nagarur, "Classification of leucocytes using convolutional neural network optimized through genetic algorithm", In: *Proc. of the 7th Annual World Conf. of the Society for Industrial and Systems Engineering*, 2018.

[14] F. A. Zeidabadi and M. Dehghani, "Poa: Puzzle optimization algorithm", *Int. J. Intell. Eng. Syst*, Vol. 15, pp. 273-281, 2022, doi: 10.22266/ijies2022.0228.25.

[15] K. Anju and K. Avanish, "An advanced approach to the employee recruitment process through genetic algorithm", *International Journal of Information Technology*, Vol. 13, pp. 313-319, 2021.

[16] S. Begum and S. S. Padmannavar, "Genetically Optimized Ensemble Classifiers for Multiclass Student Performance Prediction", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 2, 2022, doi: 10.22266/ijies2022.0430.29.

[17] R. Haldulakar and J. Agrawal, "Optimization of association rule mining through genetic algorithm", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 3, pp. 1252-1259, 2011.

[18] S. A. Muthana and K. R. K. Mahamud, "Comparison of Multi-objective Optimization Methods for Generator Maintenance Scheduling", *Methods (Multi-Objective Metaheuristics)*, Vol. 14, p. 15.

[19] J. Singh, "An Optimal Resource Provisioning Scheme Using QoS in Cloud Computing Based Upon the Dynamic Clustering and Self-Adaptive Hybrid Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 3, 2022, doi: 10.22266/ijies2022.0630.13.

[20] R. K. Sanapala and S. R. Duggirala, "An Optimized Energy Efficient Routing for Wireless Sensor Network using Improved Spider Monkey Optimization Algorithm", *Transportation*, Vol. 8, p. 9, 2022.

[21] A. M. Jubair, R. Hassan, A. H. M. Aman, H. Sallehudin, Z. G. A. Mekhlafi, B. A. Mohammed, and M. S. Alsaffar, "Optimization of Clustering in Wireless Sensor Networks: Techniques and Protocols", *Applied Sciences*, Vol. 11, p. 11448, 2021.

[22] M. H. Rangappa and G. C. Dyamanna, "Energy-Efficient Routing Protocol for Hybrid Wireless Sensor Networks Using Falcon Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol.15, No.4, 2022, doi: 10.22266/ijies2022.0831.01.

[23] P. Cinat, G. Gnecco, and M. Paggi, "Multi-scale surface roughness optimization through genetic algorithms", *Frontiers in Mechanical Engineering*, Vol. 6, p. 29, 2020.

[24] F. Rabee, Y. Liao, and M. Yang, "Minimizing multiple-priority inversion protocol in hard real time system", In: *Proc. of IEEE 11th International Conf. on Dependable, Autonomic and Secure Computing*, pp. 200-206, 2013.

[25] S. K. Dhall and C. L. Liu, "On a real-time scheduling problem", *Operations Research*, Vol. 26, pp. 127-140, 1978.

[26] C. P. I. Schärfe, R. Tremmel, M. Schwab, O.

Kohlbacher, and D. S. Marks, "Genetic variation in human drug-related genes", *Genome Medicine*, Vol. 9, pp. 1-15, 2017.

[27] E. A. King, J. W. Davis, and J. F. Degner, "Are drug targets with genetic support twice as likely to be approved? Revised estimates of the impact of genetic support for drug mechanisms on the probability of drug approval", *PLoS Genetics*, Vol. 15, p. e1008489, 2019.

[28] J. Alcaraz and C. Maroto, "A robust genetic algorithm for resource allocation in project scheduling", *Annals of Operations Research*, Vol. 102, pp. 83-109, 2001.

[29] D. O. Boyer, C. H. Martínez, and N. G. Pedrajas, "Cixl2: A crossover operator for evolutionary algorithms based on population features", *Journal of Artificial Intelligence Research*, Vol. 24, pp. 1-48, 2005.

[30] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization", *RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep,* 2013.