Integrating Exchange Market, Queen Bee, and Shuffled Complex Evolution Algorithms for Multi-variable Function Optimization

Mina Salim[1*], Sima Hamedifar[2], Aliasghar Lotfi[3]

[1,3] Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran
[2] Department of Electronics, Carleton University, Ottawa, Canada

E-mail addresses: m.salim@tabrizu.ac.ir, sima.hamedifar@gmail.com, aliasgarlotfi11@gmail.com
[*]Corresponding author

Received: 11/08/2023, Revised: 17/11/2023, Accepted:07/04/2024.

Abstract
In this paper, three popular algorithms, including the Exchange Market Algorithm (EMA), the Shuffled Complex Evolution (SCE) algorithm, and the Queen Bee (QB) algorithm, are considered to propose three new hybrid evolutionary algorithms named EMA-QB, EMA-SCE, and EMA-SCE-QB. Then, to analyze and validate the effectiveness and efficiency of these new algorithms, we compared their performance with the performance of EMA, SCE, and QB algorithms on 12 benchmark functions with 10, 20, 30, and 50 variables. It is deduced that hybridization has presented a better performance in optimum seeking from both time and accuracy points of view, which become more distinctive as the number of variables grows. Finally, the sum of run times, minimum value of cost functions, and the number of iterations obtained from the procedure of optimization of all functions using the considered algorithms are illustrated in four graphs for each number of variables, which prove the success of the proposed hybrid algorithms.

Keywords
Hybrid algorithm, Exchange market algorithm, Queen bee algorithm, Shuffled complex evolution.

1. Introduction

Optimization is the process of finding the optimum points of a target function. There are two ways to solve optimization problems: mathematical methods and meta-heuristic algorithms. Many constraints or complex target functions are so complicated that can't be solved with mathematical methods; on the other hand, metaheuristic algorithms have not any problems with constraints or complexity of equations because these methods are iteration-based and inspired by nature, these methods are straightforward and easily implementable. Nowadays, the application of meta-heuristic algorithms is common while dealing with optimization problems [1]. Their ability to find optimal solutions for complex and challenging problems in diverse fields and for various objective functions has been proven in a variety of articles [2]. The meta-heuristic algorithms can be categorized into the following five groups [3]:

1- Bio-simulated algorithms; are based on wild animals' life or the biological behavior of live creatures like grey wolf optimizer [4], artificial immune system [5], and dendritic cell algorithm [6].

2- Nature-inspired algorithms like the flower pollination algorithm [7], the bat algorithm [8], etc. inspired by natural systems [9].

3- Physics-based algorithms are inspired by natural rules of physics. For example, the gravitational search algorithm [10] is a physics-based meta-heuristic algorithm.

4- Evolutionary algorithms: these algorithms are based on the evolution of natural generations. Genetic algorithm [11] and the queen bee [12] are well-known examples of this category.

5- Swarm-based algorithms: these algorithms are based on the colony behavior of animals or insects in nature that are searching for optimum points together, and everyone wants to be nearer the swarms, which are in better positions. Particle swarm optimization [13], ant colony optimization [14] [15], and artificial bee colony algorithm (ABC) are good examples of this category.

The shuffled Complex Evolution Algorithm [16] and the exchange market algorithm [17], which we will discuss in this paper, are swarm-based evolutionary algorithms. In other words, they belong to the fourth and fifth groups.

Hybridizing these algorithms with each other or other methods presents better solutions for more complicated problems and increases accuracy, effectiveness, and efficiency [18]. Hybridization improves global and local search techniques and lessens the possibility of trapping into the local optimum in most algorithms [19, 2]. Among hybrid algorithms and their applications in the literature, the following articles can be mentioned: hybridization of GA and competitive swarm optimizer [20], hybridization of grey wolf optimizer and differential evolution [21], hybridized PSO, QB, and GA algorithms with Nelder-Mead (NM) algorithm for PID controller design for a Gryphon robot [22], hybrid honeybee mating optimization algorithm with multi-objective optimization used in searching for common patterns on protein sequences [23], an artificial bee colony based hybrid approach for waste collection problem [3], an adaptive swing-up sliding mode controller design for a natural inverted pendulum system based on the hybrid culture-bees algorithm [24], hybrid genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking [25], hybrid algorithm based on enhanced grey wolf optimization and algorithm of dragonfly for handling optimal power flow issue [26], a hybrid ant colony optimization algorithm for multi-objective vehicle routing problem with flexible time windows [27], hybridization of bee algorithm, teaching-learning-based optimization algorithm and non-dominated sorting genetic algorithm (NSGA-II) for an imperfect production system in order to increase its product quality under two warranty policies [19], hybrid multi-objective artificial bee colony for multiple sequence alignment [28]. They also have been used in multi-label classification [20], [29]; just to call a few.

In this paper, we propose three hybrid algorithms based on EMA and two other algorithms, the SCE and QB, including hybridized EMA with QB algorithm; EMA-QB, hybridized EMA with SCE method; EMA-SCE, and hybridization of all three algorithms, EMA-SCE-QB. To show the acquired benefits of hybridization, these algorithms are applied to optimize 12 benchmark functions with various dimensions of 10, 20, 30, and 50 and the obtained results are compared with the results of EMA, SCE, and QB algorithms.

The rest of the paper is organized as follows. Section 2 reviews the EMA algorithm. Section 3 and 4 reviews the QB, and the SCE methods, respectively. In Section 5, we propose our hybridized algorithms: EMA-QB, EMA-SCE, and EMA-SCE-QB. In Section 6, the benchmark functions are introduced. Simulation results are presented and discussed in section 7. Section 8 concludes the paper.

2. Exchange Market Algorithm (EMA)

The exchange market algorithm (EMA), proposed by Gorbani and Babaei [17], is formed based on a flash of inspiration from shareholders' performance in the stock market during the procedure of trading the shares. This novel optimization method is appropriate for both continuous and discrete [30] optimization problems and , compared with other metaheuristics, has exhibited superb capability of finding the global optimal point, more robustness and efficiency and even fewer limitations like trapping into local optimum points, early convergence, disability in finding adjacent points of the optimal point, converging to different solutions in each implementation of the program. Furthermore, this algorithm was utilized in economic emission dispatch and reliability in thermal power plant [31] to reduce fuel and emission costs of the system and raise the dependability.

This algorithm, simulates the competitive behavior of shareholders in buying and selling their shares in the virtual stock market in order to elevate their ranking up to the list of successful shareholders. In this algorithm two general states are considered: the normal state and the oscillation state. First, the shareholders' population is divided into three groups according to their share amount. The first group, which usually forms 10-30 percent of the total population, is shareholders with the highest rank, while the second group includes 20-50 percent of middle-ranked exchange market dealers, and the rest of the members with the lowest shares are classified in the third group. People in the first group don't trade their shares in both states. People in the second group, in the normal state, try to search around the optimum point using the differences in the amounts of the shares of the first group's members. In the oscillation state, they try to find solutions with minimum cost value with the help of buying and selling their shares in such a way that their total share amount remains constant. People in the third group, take more risk than those in the second group, and in the normal state, they use the differences in shares between their individuals and the first group's individuals and the differences in shares of the first group's members. In the oscillation state, they try to find solutions with minimum cost value by changing their total share amount. The following steps summarize the EMA algorithm.

step 1. Initialize the population of shareholders and evaluate them. Do the following steps for $k = 1, \cdots, iter_{max}$.

step 2. Sort the population according to their costs.

step 3. Divide the sorted population into three groups according to their rank.

step 4. Perform the normal state.

a) For jth member of the second group, $pop_j^{group(2)}$, change the shares by

$$pop_j^{group(2)} = r \times pop_{1;i}^{group(1)} + (1-r) \times pop_{2;i}^{group(1)} \tag{1}$$

where $pop_{1;i}^{group(1)}$ and $pop_{2;i}^{group(1)}$ are two randomly selected members of the first group, r is a random value $(0 < r < 1)$, j=1,2,…,n_j, and n_j is the number the second group members.

b) For kth member of the third group, $pop_k^{group(3)}$, do:

i. Define the share variations s_k by

$$s_k = 2 \times r_1 \times \left(pop_{1;i}^{group(1)} - pop_k^{group(3)} \right) + 2 \times r_1 \times \left(pop_{2;i}^{group(1)} - pop_k^{group(3)} \right) \tag{2}$$

where r_1 is a random value $(0 < r_1 < 1)$.

ii. Replace $pop_k^{group(3)}$ with the new member $pop_k^{group(3),new}$ according to

$$pop_k^{group(3),new} = pop_k^{group(3)} + 0.8 \times s_k \tag{3}$$

step 5. Perform the oscillation state

a) Do the following steps for all members of the second group

i. Determine the amount of shares (Δn_{t1}) that should be randomly added to some shares by

$$\Delta n_{t1} = n_{t1} - \delta + \left(2 \times r \times \mu \times \eta_1\right). \tag{4}$$

n_{t1} is total share of t th member before applying the share changes, δ is the information of the exchange market, r is a random value $(0 < r < 1)$, and μ is the ratio of the number of tth member to the number of the last member of the population. η_1 is the risk level of each member in the second group and can be determined as follows:

$$\eta_1 = n_{t1} \times g_1 \tag{5}$$

$$g_1^k = g_{1,max} - \frac{g_{1,max} - g_{1,min}}{iter_{max}} \times k \tag{6}$$

where k is the current iteration number, and $g_{1,max}$ and $g_{1,min}$ are the maximum and minimum risk values, respectively.

ii. Determine the amount of shares (Δn_{t2}) that should be sold by

$$\Delta n_{t2} = n_{t2} - \delta \tag{7}$$

where n_{t2} is the share amount of tth member after applying the step 5-a-i.

b) Do the following steps for all members of the third group.

i. Determine the amount of shareshat t (Δn_{t3}) should be randomly added to the share of each member by

$$\Delta n_{t3} = 4 \times r_s \times \mu \times \eta_2 \tag{8}$$

where r_s is a random value $(-0.5 < r_s < 0.5)$, η_2 is the risk level of each member in the third group and can be determined by

$$\eta_2 = n_{t1} \times g_2 \tag{9}$$

, and g_2 is obtained in the same way as g_1.

Remark1: In this algorithm, indeed, in normal mode, the crossover phase and in the oscillation mode, the mutation phase is performed for the second and the third group.

3. The Queen Bee Algorithm

The mating process of honeybees and their intelligent behavior while searching for food supplies have been a source of inspiration to come up with one of the evolutionary algorithms called bee algorithm, which first introduced in 1997 by Sato and Hagiwara. Since then, many modifications have been made to expand its applications in different optimization problems such as traveling salesman scheduling and constraint problems, the quadratic assignment problem (QAP) [32], distributed consensus tracking of unknown nonlinear chaotic delayed fractional-order multi-agent systems [33], and converter designing in power electronics [34]. This algorithm is studied in three various fields: the queen bee (QB), the artificial bee colony, and the fast marriage in honey bee optimization (FMBO), and it's modified version, i.e., the modified fast marriage in honey bee optimization (MFMBO) [35]. The QB algorithm, presented by Jung in 2003, is a collective search algorithm and has similarities with the genetic algorithms. It has been applied in various fields, including multivariable problems, neural networks training and feed-forward neural networks, and navigation.

In this algorithm, the population is arranged in ascending order of cost function and the queen, as one of the parents, is the best member of the group. Indeed, in the crossover step, unlike the genetic algorithm, one of the parents is always the queen. Then, the mutation phase with two unlike mutation rates is implemented in such a way that the normal mutation rate with the probability p_m is applied to better members of the population $(\xi\%)$, and the strong mutation rate with $\dot{p}_m$ probability $(\dot{p}_m > p_m)$ is used to other members $((1-\xi)\%)$ to avoid early convergence [36].

4. The Shuffled Complex Evolution Algorithm

Initially proposed by Duan et al. in 1992, the shuffled complex evolution (SCE) algorithm is derived from sharing information in natural biological evolution [37]. Since then, this global search methodology has been modified and applied to solve various optimization problems [38, 39]. SCE is a swarm-based evolutionary algorithm. Like other swarm-based algorithms, in this algorithm, initial population is evaluated in every iteration. They are ranked based on the objective function of each individual, but the point is in the complexes. Clustering the individuals into complexes causes to search be more efficient because in the first complex, the responses are better, and the offspring from them, can find optimum points better than these points; on the other hand, by this method, the random search ability is not missed, so SCE prevent getting stuck into local optimum points. The steps of optimizing a cost function (f) by the SCE algorithm are as follows:

step 1. Considering N_{pop} random points in the search space.

step 2. Calculation of the cost function for each member and putting them in ascending order.

step 3. Dividing the population into p complexes.

step 4. Evolving each complex:

a) Choose q random points with larger probabilities and smaller cost values from each complex to form a sub-complex.

b) Omit the worst point (x_w) in each sub-complex, calculate the average of other points $(\bar{x})$, and reflect x_w with respect to $\bar{x}$ by $x_r = 2\bar{x} - x_w$.

c) If $f_r < f_w$, replace x_w by x_r and go to step e, else calculate contraction point by $\left(x_c + \bar{x}\right)/2$.

d) If $f_c < f_w$, replace x_w by x_c, else go to step e.

e) Generate a random point in the search space and replace x_w with it.

f) Repeat steps b to g, r times $(r \geq 1)$.

g) Repeat steps a to g, s times $(s \geq 1)$.

step 5. Check the stopping conditions, if not fulfilled go to step 4.

5. Hybridizing algorithms

As we mentioned before, metaheuristic algorithms are growing by researchers, and one crucial way to grow and make their performance better, is hybridizing them [40]. Already, there are many hybrid algorithms like GA-PSO [41], GA-ACO [42], and SFLA-CSO[43] that make the speed and accuracy of optimization better in comparison with the old algorithms.

In summary, the advantages of hybridization on the subject of biology and genetics can be enlisted as:

I. Hybrids can combine desirable features from parent specious, so that the overall performance can improve.

II. Hybridization can reduce the adverse effects of inbreeding, leading to genetic diversity and healthier populations.

III. Combining different genetic algorithm variants can balance exploration and exploitation; as a result, it leads to more effective optimization.

IV. Hybridization helps avoid the problem of early convergence, where genetic algorithms end up with suboptimal solutions.

V. Integrating heuristic optimization methods alongside genetic algorithms enhances the global search ability.

VI. Combining algorithms with complementary strengths can speed up the convergence.

In this paper, we hybridize the EMA algorithm with the QB, and SCE and introduce EMA-QB and EMA-SCE. In addition, we hybridize the EMA with both the QB and the SCE and introduce the EMA-SCE-QB algorithm. The way that we hybridize algorithms is essential for improving convergence speed, and accuracy. The results prove the superiority of these hybridized algorithms compared to their basic algorithm in terms of accuracy, consistency, and convergence speed.

5.1. Hybrid EMA-QB Algorithm

To enhance the features of the exchange market algorithm in function optimization, it is hybridized with the queen bee algorithm. As we know, a modification is made within the queen bee algorithm with the goal of improving the genetic algorithm, where the crossover step involves the random selection of one parent. In contrast, the “queen”, symbolizing the best member, is held constant throughout the algorithm.

As mentioned in Remark 1, the exchange market algorithm in normal mode has a crossover performance on the members of the second and third groups. Therefore, two members of the first group are randomly selected. This motivated us to hybridize the EMA and the QB algorithm. In this new algorithm, only one of these two members is chosen randomly, and the second one is the first member of the first group, which has the lowest cost as a queen in the QB algorithm. To avoid early convergence, a more significant mutation rate is considered in the QB algorithm. As mentioned in Remark1, the EMA performs mutation on the members of the second and third groups in the oscillation mode. According to step 5-a-i, the mutation rate in the second group is small, but according to step 5-b-i, the mutation rate in the third group is big enough to avoid premature convergence. Therefore, the hybridization should be done just for the third group. We hybridized the bee algorithm and the EMA for both the second, and the third group and the results confirmed this point. Hence, we eliminated the results for the hybridization of the second group. Consequently, in the EMA-QB algorithm the formula in step 4-b-i changes to

$$\begin{aligned} s_k = {} & 2 \times r_1 \times \left(pop^{QB} - pop_k^{group(3)} \right) + \\ & 2 \times r_1 \times \left(pop_{2;i}^{group(1)} - pop_k^{group(3)} \right) \end{aligned} \tag{10}$$

Where pop^{QB} is the first member of the first group $(pop^{QB} = pop_{1;1}^{group(1)})$. The flowchart of the EMA-QB is Fig. 1.

5.2. Hybrid EMA-SCE Algorithm

To improve the features of the EMA algorithm, it can also be hybridized with the SCE algorithm. As we see in Section 4, grouping individuals into complexes enhances the efficiency of the search process. This is due to the fact that the first complex tends to yield superior responses, and offspring generated from this complex have a better chance of finding optimal points compared to others. Conversely, within the EMA framework, the initial group comprises the top-performing individuals, eliminating the necessity to apply to the SCE algorithm to improve this group. To this end, one choice is to change the distribution of the shareholders’ populations like the SCE algorithm. After sorting the population of shareholders according to their share amount, 10-30 percent of top shareholders are selected for the first group, just like the EMA algorithm. The hybridization with the SCE algorithm is performed for the second and third group. For the rest of the members, distribution of shareholders is like distribution of members in the SCE algorithm. The first member will be placed in the second group, the second member will be placed in the third group, and the third one will be placed in the second group; this routine continues until all members are placed in the second and the third group. By this method, the distribution of the members will be uniform. This kind of distribution is performed for the second and the third group in both normal and oscillation states. It is also possible that the number of the members in these two groups is not the same; as long as they are distributed in the described manner, it can be called the EMA-SCE algorithm. The flowchart of the EMA-SCE is depicted in Fig. 2.

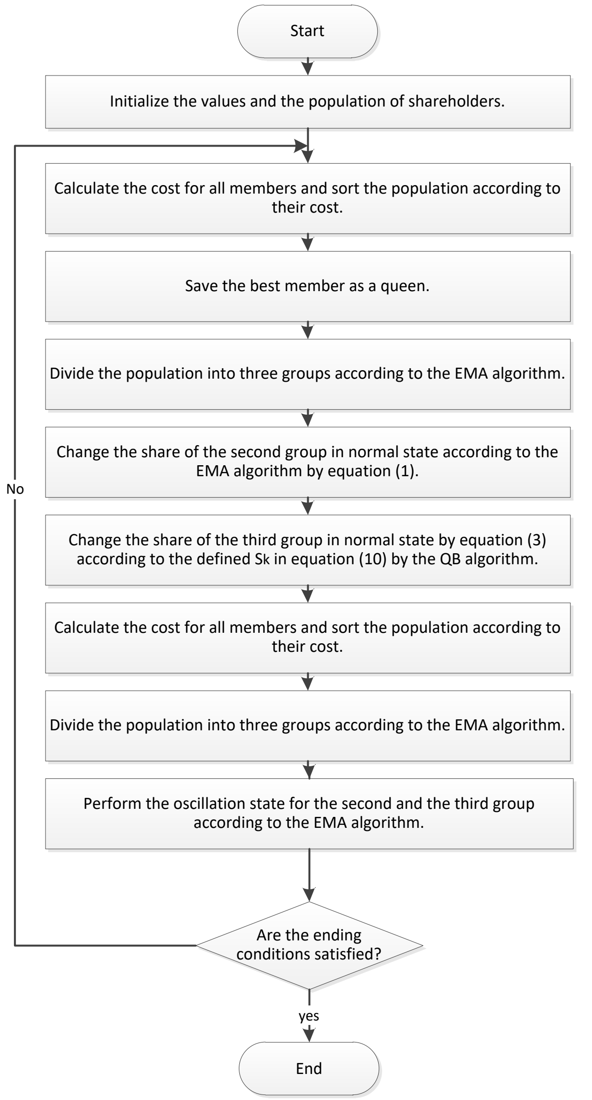


Fig. 1. The flowchart of EMA-QB algorithm

5.3. *Hybrid EMA-SCE-QB Algorithm*

To improve the crossover in the EMA-SCE algorithm, it can be hybridized again with the QB algorithm. This means that in the third group of the EMA-SCE, the crossover is performed similarly way with the QB algorithm, i.e., crossover with one fixed parent, which is the queen bee. However, in the second group, the crossover is performed like EMA and EMA-SCE algorithms. Therefore, the obtained algorithm has two main differences from the EMA algorithm. First, the population distribution in the second and in third groups is similar to the SCE algorithm in both the normal state and oscillation state. Second, the crossover in the third group is like the QB algorithm. The flowchart of EMA-SCE-QB is illustrated in Fig. 3.

Comparing the speed of different evolutionary algorithms in finding the optimum point of a function can be done through empirical experiments and performance metrics. Here's a general approach [44].

I. Select a set of benchmark functions that are commonly used for testing optimization algorithms.

II. Ensure the selected algorithms have the same parameters and configuration or similar settings to make the comparison fair.

III. Define performance metrics to evaluate the algorithms. Typical metrics include convergence speed, solution quality, and robustness.

IV. Run each algorithm on the selected benchmark function, collecting data on how they perform. Multiple runs can be conducted to account for randomness.

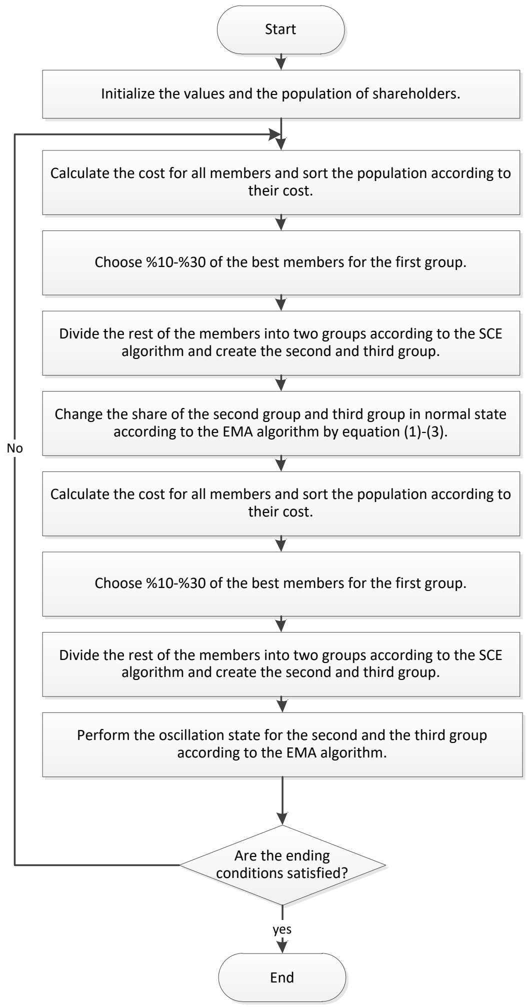


Fig. 2. The flowchart of EMA-SCE algorithm

Comparing the complexity of different evolutionary algorithms for function optimization involves assessing various aspects of the algorithms, such as population size, number of generations, and parameter sensitivity. Also, there is a trade-off between solution quality and computational time that should be considered [45].

6. The Benchmark Functions

The functions of Table I. are employed to perform simulations and test the proposed methods. The optimal function value in all the functions is zero. All the applied functions are high dimensional. About the number of local minimums or modalities, Ackley, Griewank, Penalized functions 1 and 2 and Rastrigin functions are multimodal. However, Quartic, Rosenbrock, and Schwefel's functions

1.2, 2.21, and 2.22, Sphere and Step functions are unimodal. On the other hand, about separability, Ackley, Griewank, Penalized functions 1 and 2, Rosenbrock, Schwefel's function 1.2, 2.21, and 2.22 are non-separable which are consequently difficult to optimize, but Quartic, Rastrigin, Sphere, and Step functions are separable and optimization of these functions becomes problematic if they are multimodal.

Each function has a particular property that makes its optimization difficult. For instance, the Ackley function has an exponential term that covers its surface with numerous local minimums. To obtain good results for this function, the searching strategy must combine the components of exploring and exploiting efficiently. The Griewank function has a product term that introduces interdependence among the variables. Therefore, the techniques that optimize each variable independently experience a failure. As in the Ackley function, the optima of the Griewangk function are regularly distributed. The contour in the Rastrigin function has a great deal of local minimums that are regularly distributed, and their value increases with the distance to the global minimum. A significant problem with this function is the fact that an optimization algorithm could easily trap in a local minimum.

7. Numerical Results

The performance of proposed algorithms, including EMA-SCE, EMA-QB, and EMA-QB-SCE, is evaluated by implementing them on the benchmark functions mentioned in the previous section. For comparison, the results of EMA, SCE and QB algorithms are considered as well. Execution time and minimum of cost functions while implementing the mentioned algorithms for 10, 20, 30, and 50 variables are illustrated in Tables I-IX, respectively.

According to the results in Tables II-V, overall, the speed of hybrid algorithms is the highest in the optimization of the most functions, especially, when the number of variables rises; they present better results in almost all the functions. The difference in run time between hybrid algorithms, the QB, and the SCE algorithms, having the most amount of run time, is considerably significant except for a small number of functions and specific number of variables. However, the hybridization of EMA with these algorithms has resulted in far less run time in EMA-SCE-QB, EMA-QB, and EMA-SCE algorithms than QB and SCE algorithms, resulting in an improvement from the run time point of view. Minimums of cost functions after employing each algorithm are depicted in Tables V-IX.

First of all, it is crystal clear that, generally, an algorithm may be successful in the optimization of one function and unsuccessful in finding the optimum point for another function. However, the obtained results illustrate that the number of accomplishments of hybrid algorithms in reaching the global optimum point with the least error is much more than other algorithms.

According to Table II, the execution time of optimization of functions with 10 variables is the least when the process is performed by hybrid algorithms, including EMA-QB, for most of the functions such as f_2, f_3, f_4, f_5, f_9, f_{10}, and f_{11}, and EMA-QB-SCE for f_7. EMA- QB is the second fast algorithm for f_1 and f_{12} with, respectively, $7.7ms$ and $3.7ms$ differences with EMA, and for f_8 being $54ms$ later than SCE. In f_6, EMA-SCE is the second fastest algorithm, being $14ms$ slower than EMA. However, QB and SCE (except for f_8) are the slowest algorithms, with an average run time of $10.22s$ and $5.35s$.

For 20 variables, the proposed algorithms are the most appropriate algorithms from run time point of view, including EMA-SCE-QB for f_1, and f_4, EMA-SCE for f_2, f_7, and f_{10}, and EMA-QB for f_5, f_8, f_9, and f_{11}, concerning Table 3.

Although EMA had the least run time for the remaining functions, hybrid algorithms EMA-QB for f_3 and EMA-SCE for f_6 and f_{12} are only $0.1ms$, $54ms$, and $1.6ms$ slower. The QB and SCE algorithms are the slowest with minimum $0.43s$ and $0.203s$ and maximum $31.57s$ and $46.45s$ run time.

In state of 30 variables, as illustrated in Table 4, hybrid algorithms performed more desirably. They consumed less time in comparison to other algorithms in a way that EMA-SCE in case of f_1, f_2, f_6, f_7, f_{10}, f_{11}, and f_{12}, EMA-QB in case of f_3, f_5, and f_9, and EMA-SCE-QB in case of f_4 and f_8 optimized fast while the average run time of EMA, QB, and SCE is 0.95, 15.95, and 36.96 seconds, respectively.

Table V depicts that, for the dimension of 50, the functions f_1, f_2, f_3, f_4, f_{10}, and f_{12} are optimized at the shortest run time by EMA-SCE, when for f_5, f_7, and f_9 it is EMA-QB and for f_8 it is EMA-SCE-QB. On the other hand, EMA, QB, and SCE are carried out at 3.41, 37.79, and 18.39 seconds, respectively.

For the data illustrated in Tables VI-IX, the accuracy of algorithms in optimizing each function is investigated separately to know which algorithm optimizes a function with more precision.

As mentioned previously, the Ackley function (f_1) is challenging to optimize due to its multimodality and abundance of local optimum points. Nevertheless, EMA-SCE for 50 variables and EMA-SCE-QB for 20 and 30 variables performed better than other algorithms with the least average error.

While evaluating the Griewank function (f_2), from the perspective of least error, hybrid algorithms have expressed their superiority as the number of variables increases. For example, the disparity between the minimums of cost functions optimized by EMA and hybrid algorithms increases to an ignorable amount when comparing EMA with EMA-SCE-QB for $D = 20$ and to $0.3E - 6$ when comparing to EMA-SCE for D=50. This is while the errors of the QB and the SCE algorithms are equal to $2.7E - 4$ and 4.305.

In optimizing Penalized function 2 (f_4), the EMA-QB algorithm has carried out optimization more precisely for dimensions 20, 30, and 50 with an average error equal to $6.97E - 6$ and for dimensions 10, EMA-SCE-QB performed with less error than other algorithms except the SCE algorithm where there is $0.4E - 6$ difference between their values.

While dealing with the optimization of the Quartic function (f_5), the SCE algorithm for 10 variables and the QB algorithm for the other dimensions have optimized this function better.

Table I: benchmark functions

Function	Formulation	Range
1.Ackley	$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32,32]$
2.Griewank	$f_2(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]$
3.Penalized function 1	$f_3(x) = \frac{\pi}{n}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1 + 10\,sin^2(\pi y_{i+1})] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i = 1 + {x_i+1}/_4$ $u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i-a)^m x_i < -a \end{cases}$	$[-50,50]$
4.Penalized function 2	$f_4(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i-1)^2[1+ sin^2(3\pi x_{i+1})] + (x_n-1)^2[1+ sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i,5,100,4)$	$[-50,50]$
5.Quartic	$f_5(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	$[-1.28,1.28]$
6.Rastrigin	$f_6(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]$
7.Raosenbrock	$f_7(x) = \sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2 + (x_i-1)^2]$	$[-30,30]$
8.Schwefel's function 1.2	$f_8(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	$[-100,100]$
9.Schwefel's function 2.21	$f_9(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	$[-100,100]$
10.Schwefel's function 2.22	$f_{10}(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]$
11.Sphere	$f_{11}(x) = \sum_{i=1}^{n} x_i^2$	$[-100,100]$
12.Step	$f_{12}(x) = \sum_{i=1}^{n}(\lfloor x_i + 0.5\rfloor)^2$	$[-200,200]$

Table II. run time for 10 variables

	EMA	**SCE**	**QB**	**EMA-QB**	**EMA-SCE**	**EMA-SCE-QB**
f1	0.043	0.225	17.009	0.051	0.054	0.056
f2	0.038	0.120	5.529	0.038	0.038	0.039
f3	0.031	0.083	0.276	0.029	0.033	0.032
f4	0.046	0.161	9.521	0.045	0.049	0.047
f5	0.034	0.114	0.135	0.030	0.037	0.033
f6	0.142	0.238	17.52	0.165	0.156	0.232
f7	0.302	35.39	15.27	0.287	0.199	0.193
f8	0.404	0.212	24.52	0.266	0.437	0.281
f9	0.074	22.16	16.92	0.066	0.084	0.074
f10	0.042	0.254	17.041	0.041	0.042	0.042
f11	0.033	0.145	16.048	0.031	0.034	0.034
f12	0.026	0.076	0.420	0.030	0.042	0.040

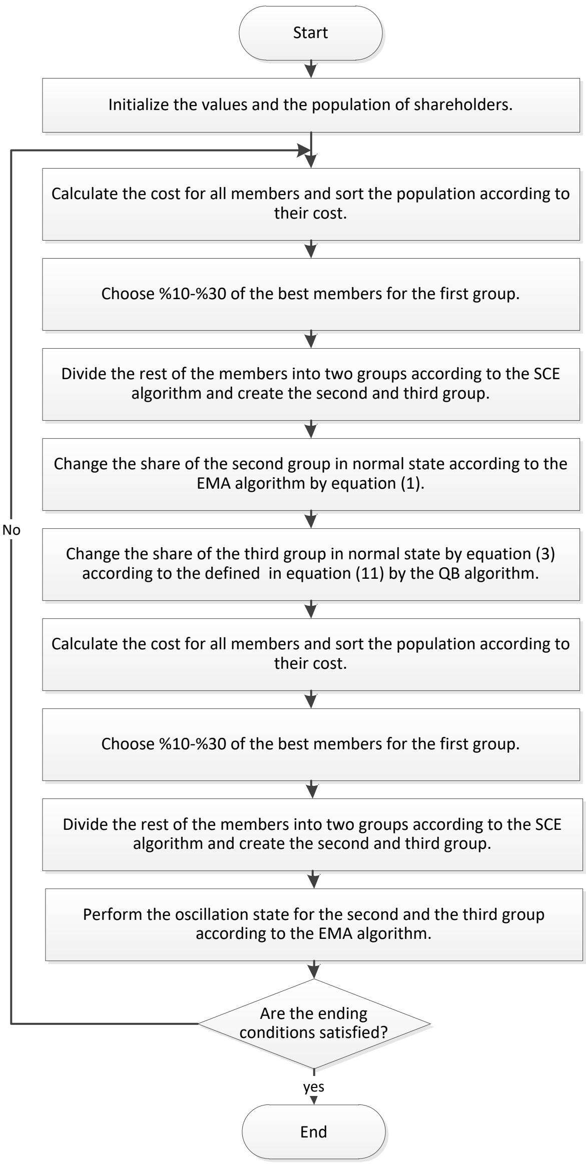


Fig. 3. The flowchart of EMA-SCE-QB algorithm

However, with less difference, hybrid algorithms are ranked after them. Considering the average error of all algorithms in the case of the Rastrigin function (f_6), EMA-SCE has the slightest average error, and for 50 variables, EMA-SCE-QB expresses better performance.

The Raosenbrock function (f_7) and the Schwefel's function 2.21 (f_9) are optimized best by hybrid algorithms for all dimensions. In the case of f_7, EMA-SCE-QB is the most accurate algorithm for $D = 10$ and 20 with the error of $9.48E-6$ and $9.18E-6$, and EMA-SCE has the slightest error of $9.53E-6$ and 2.26 for $D = 30$ and 50; meanwhile, other hybrid algorithms like EMA-SCE-QB with the difference of $0.12E-6$ for $D = 30$ and 0.3 for $D = 50$, have relatively close results to the mentioned algorithms. The QB and SCE algorithms indicate the weakest strength in this process.

The error of EMA is $8.87E-6$ for $= 10$, rising to 6.05 for $D = 50$. In the case of f_9, EMA-QB has an average error of $9.72E-6$ while it is $9.75E-6$ for EMA and 0.58 and 3.6 for the QB and SCE algorithms. The error of

EMA is close to the error of EMA-QB, and it is equal to the error of other hybrid algorithms, but the difference in errors of the QB and the SCE algorithms is considerable.
In the case of Schwefel's function 1.2 (f_8), although SCE is more effective in optimizing the function with 10 variables, when the number of variables goes up, hybrid algorithms indicate better performance. For instance, for the dimension of 50, EMA-QB, EMA-SCE-QB, and EMA-SCE have, respectively, the errors of $9.93E-6$, $9.94E-6$, and $9.96E-6$ while the SCE algorithm results in the error of 976.96.
Schwefel's function 2.22 (f_{10}) is better optimized by EMA-QB for $D=10$ and 20, EMA-SCE for $D=30$, and EMA for $D=50$. Considering the average error of these algorithms, which are equal to $9.30E-6$, $9.38E-6$, and $9.83E-6$, it is inferred that, in totally, hybrid algorithms had better performance than EMA.

Table III. run time for 20 variables

	EMA	SCE	QB	EMA-QB	EMA-SCE	EMA-SCE-QB
f1	0.112	37.158	17.215	0.118	0.110	0.109
f2	0.07	37.886	16.216	0.076	0.067	0.069
f3	0.064	14.722	2.495	0.064	0.066	0.068
f4	0.138	37.132	17.133	0.152	0.138	0.137
f5	0.084	0.203	0.430	0.077	0.087	0.079
f6	0.335	37.428	18.091	0.405	0.389	0.425
f7	1.301	37.019	16.999	1.183	0.775	0.839
f8	2.181	45.63	31.578	1.467	2.288	1.468
f9	0.227	43.067	17.091	0.201	0.243	0.242
f10	0.087	38.976	17.188	0.083	0.077	0.077
f11	0.059	37.048	16.930	0.058	0.060	0.060
f12	0.061	46.455	3.227	0.066	0.062	0.063

Table IV. run time for 30 variables

	EMA	SCE	QB	EMA-QB	EMA-SCE	EMA-SCE-QB
f1	0.190	37.306	17.335	0.199	0.187	0.193
f2	0.123	38.468	19.168	0.128	0.106	0.109
f3	0.111	38.460	4.556	0.108	0.115	0.114
f4	0.338	37.264	17.593	0.391	0.297	0.294
f5	0.176	0.373	1.539	0.150	0.183	0.152
f6	0.686	37.487	18.254	0.743	0.681	0.699
f7	2.578	37.036	17.312	2.640	1.807	1.838
f8	6.418	49.796	38.519	4.461	6.708	4.293
f9	0.491	43.740	17.283	0.444	0.515	0.514
f10	0.145	38.390	17.342	0.146	0.117	0.127
f11	0.093	37.101	17.336	0.095	0.089	0.093
f12	0.105	48.160	5.299	0.091	0.086	0.092

Table V. run time for 50 variables

	EMA	SCE	QB	EMA-QB	EMA-SCE	EMA-SCE-QB
f1	0.468	37.792	17.541	0.508	0.456	0.468
f2	0.272	38.602	19.646	0.284	0.211	0.217
f3	0.222	38.636	8.941	0.223	0.219	0.221
f4	1.124	37.483	18.020	1.231	0.764	0.815
f5	0.497	0.870	4.273	0.405	0.512	0.408
f6	1.399	37.524	18.431	1.526	1.470	1.627
f7	4.239	37.363	17.751	4.234	4.353	4.391
f8	30.450	58.403	52.996	22.797	31.960	20.020
f9	1.398	43.828	17.425	1.240	1.419	1.426
f10	0.353	37.978	17.627	0.340	0.245	0.259
f11	0.185	37.113	17.457	0.187	0.164	0.176
f12	0.278	47.925	10.592	0.224	0.168	0.172

Table VI. minimum of cost function for 10 variables

	EMA	SCE	QB	EMA-QB	EMA-SCE	EMA-SCE-QB
f1	9.00E-06	8.64E-06	0.004	8.80E-06	8.88E-06	8.99E-06
f2	8.51E-06	7.39E-06	5.20E-06	8.34E-06	8.17E-06	8.14E-06
f3	8.19E-06	7.35E-06	7.08E-06	8.02E-06	7.95E-06	8.13E-06
f4	8.33E-06	7.33E-06	1.79E-05	8.16E-06	8.27E-06	7.72E-06
f5	7.41E-06	7.12E-06	7.93E-06	7.73E-06	7.50E-06	7.19E-06
f6	7.99E-06	7.98E-06	9.68E-05	8.35E-06	7.78E-06	8.34E-06
f7	8.87E-06	1.326	0.0002	9.13E-06	8.94E-06	8.48E-06
f8	9.29E-06	7.34E-06	0.071	9.06E-06	9.43E-06	9.15E-06
f9	9.43E-06	0.00019	0.040	9.39E-06	9.46E-06	9.45E-06
f10	9.39E-06	8.59E-06	0.001	8.50E-06	8.79E-06	8.83E-06
f11	8.01E-06	7.53E-06	0.0001	8.39E-06	7.86E-06	8.32E-06
f12	0	0	0	0	0	0

Table VII. minimum of cost function for 20 variables

	EMA	SCE	QB	EMA-QB	EMA-SCE	EMA-SCE-QB
f1	9.29E-06	1.515	0.0107	9.48E-06	9.29E-06	9.58E-06
f2	9.22E-06	0.0008	1.57E-05	9.29E-06	9.22E-06	9.21E-06
f3	8.93E-06	1.58E-05	7.58E-06	9.00E-06	8.93E-06	9.28E-06
f4	9.23E-06	0.099	0.0001	7.67E-06	9.23E-06	8.31E-06
f5	8.55E-06	8.07E-06	6.73E-06	8.84E-06	8.55E-06	8.59E-06
f6	8.88E-06	38.515	37.867	9.20E-06	8.88E-06	9.04E-06
f7	9.46E-06	4.750	14.495	9.45E-06	9.46E-06	9.18E-06
f8	9.84E-06	9.145	8.862	9.71E-06	9.84E-06	9.78E-06
f9	9.77E-06	2.439	0.261	9.75E-06	9.77E-06	9.76E-06
f10	9.45E-06	0.124	0.004	9.33E-06	9.45E-06	9.39E-06
f11	8.91E-06	0.194	0.001	8.96E-06	8.91E-06	8.98E-06
f12	0	5.3	0	0	0	0

Hybrid algorithms express their strong ability in optimization of the Sphere function (f_{11}) compared to other algorithms, especially, when the number of variables is equal to 30. In this situation, EMA–QB ends in a result equal to $9.27E-6$ while the result of EMA is $9.39E-6$ the QB and SCE algorithms have significantly large amounts of 0.0056 and 22.63 in comparison.
The Step function (f_{12}) is optimized without any error by all the algorithms for all numbers of variables. However, by growth of dimension, the QB algorithm and, especially, the SCE algorithm show weakness in optimization of this function such that for $D=30$ and 50, the error of the SCE algorithm is 48 and 631.4, respectively, and the error of the QB algorithm for $D=50$ is 0.

Table VIII. minimum of cost function for 30 variables

	EMA	SCE	QB	EMA-QB	EMA-SCE	EMA-SCE-QB
f1	9.57E-06	3.21	0.017	9.69E-06	9.68E-06	9.53E-06
f2	9.52E-06	0.296	4.52E-05	9.25E-06	9.26E-06	9.44E-06
f3	9.19E-06	0.001	7.69E-06	9.44E-06	9.45E-06	9.28E-06
f4	9.29E-06	0.8612	0.001	6.57E-06	8.88E-06	8.19E-06
f5	9.13E-06	9.24E-06	6.59E-06	9.07E-06	9.06E-06	9.04E-06
f6	9.31E-06	672.115	234.60	9.34E-06	9.38E-06	9.51E-06
f7	0.331666	24.913	42.862	0.138014	9.53E-06	9.65E-06
f8	9.90E-06	135.31	98.200	9.87E-06	9.86E-06	9.80E-06
f9	9.87E-06	5.464	0.540	9.82E-06	9.87E-06	9.87E-06
f10	9.58E-06	1.363	0.011	9.56E-06	9.49E-06	9.51E-06
f11	9.39E-06	22.630	0.005	9.27E-06	9.50E-06	9.50E-06
f12	0	47.93	0	0	0	0

Table IX. minimum of cost function for 50 variables

	EMA	SCE	QB	EMA-QB	EMA-SCE	EMA-SCE-QB
f1	9.81E-06	5.4701	0.033	9.78E-06	9.76E-06	9.79E-06
f2	9.72E-06	4.305	0.0002	9.67E-06	9.42E-06	9.58E-06
f3	9.65E-06	0.0290	8.50E-06	9.75E-06	9.60E-06	9.75E-06
f4	9.33E-06	19.396	0.0019	6.67E-06	9.12E-06	9.15E-06
f5	9.42E-06	9.57E-06	6.82E-06	9.51E-06	9.45E-06	9.47E-06
f6	9.64E-06	20344.924	1250.0071	9.64E-06	9.70E-06	9.60E-06
f7	6.051	95.868	96.001	6.343	2.256	2.559
f8	9.96E-06	976.691	1421.93	9.93E-06	9.96E-06	9.94E-06
f9	9.93E-06	8.834	1.509	9.93E-06	9.94E-06	9.93E-06
f10	9.74E-06	7.484	0.042	9.81E-06	9.82E-06	9.81E-06
f11	9.51E-06	397.133	0.023	9.68E-06	9.56E-06	9.69E-06
f12	0	631.4	0.066	0	0	0

9. References

[1] X.-S. Yang, "*Nature-Inspired Optimization Algorithms*", London: Elsevier, 2014.

[2] Q. Wei, Z. Guo, H. C. Lau, and Z. He, "An artificial bee colony-based hybrid approach for waste collection problem with midway disposal pattern," *Appl. Soft Comput.*, vol. 76, pp. 629–637, 2019.

[3] A. Kumar and S. Bawa, "A comparative review of meta-heuristic approaches to optimize the SLA violation costs for dynamic execution of cloud services", *Soft Compute*, vol. 24, pp. 3909–3922, 2020.

[4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "A Grey wolf optimizer," *Adv Eng Softw*, vol. 69, pp. 46–61, 2014.

[5] J.O. Kephart, "A biologically inspired immune system for computers", *In: Artificial life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems*, pp. 130–139, 1994.

[6] J. Greensmith and U. Aickelin, "The deterministic dendritic cell algorithm," *SSRN Electron. J.*, 2008.

8. Conclusion

In this paper, we presented three novel hybrid algorithms i.e., EMA-QB, EMA-SCE, and EMA-SCE-QB, based on three prominent and popular algorithms named EMA, SCE, and QB. To prove the effectiveness of the proposed algorithms, we implemented them for finding the optimum point of 12 continuous functions. We compared the obtained results with the ones from basic algorithms. By analyzing the obtained results, it was concluded that hybridization has resulted in enhancement in EMA algorithm procedure and formation of competitive algorithms with advantages like high accuracy, consistency, speed , and less execution time. In the future, we intend to utilize these reliable algorithms in the control area to design the transfer functions of various controllers for diverse plants and systems like robotic systems.

[7] M. Yang and X. He, "Flower pollination algorithm: a novel approach for multi objective optimization," *Eng Optim*, vol. 46, no. 9, pp. 1222–1237, 2014.

[8] XS. Yang ," Bat algorithm for multi-objective optimization", *Int J Bio Inspir Comput* , vol. 3, no. 5, pp.267–274, 2011.

[9] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *arXiv [cs.NE]*, 2013.

[10] Ll. Yang, Wy. Qian, Q. Zhang ," Central force optimization", *J Bohai Univ (Natural Science Edition)* , vol. 32, no. 3, pp. 203–206, 2011.

[11] J. Holland ," Adoption in natural and artificial systems", *University of Michigan Press*, Michigan, 1975.

[12] S. H. Jung, "Queen-bee evolution for genetic algorithms", *Electron Lett*, vol. 39, no. 6, pp. 575–576, 2003.

[13] J. Kennedy ," Particle swarm optimization", *In: Sammut C, Webb GI (eds) Encyclopedia of machine learning*, Springer, Boston, MA, 2011.

[14] M. Dorigo and C. Blum, "Ant colony optimization theory: a survey", *Theor Comput Sci*, vol. 344, no. 2–3, pp. 243–278, 2005.

[15] M. Dorigo and D. Caro, “Ant colony optimization: a new meta-heuristic”, in *Proceedings of the 1999 congress on evolutionary computation-CEC 99*, vol. 2, IEEE, 1999, pp. 1470–1477.

[16] Q. Duan, V.K. Gupta, S. Sorooshian, " A shuffled complex evolution approach for effective and efficient global theory minimization", *Journal of optimization and applications*, vol.76, no. 3, pp. 501-521, 1993.

[17] N. Ghorbani, E. Babaei, "Exchange Market Algorithm," *Applied Soft Computing*, vol. 19, pp. 177-187, 2014.

[18] A. A. Taleizadeh, P. Pourrezaei Khalegi, I. Moon, "Hybrid NSGA-II for an imperfect production system considering product quality and returns under two warrenty policies," *Applied Soft Computing Journal*, vol. 75, pp. 333-348, 2018.

[19] Z. Zhang, S. Ding and W. Jia, "A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 254-268, 2019.

[20] Y. Ding, K. Zhou and W. Bi, "Feature selection based on hybridization of genetic algorithm and competitive swarm optimizater", *Soft Computing*, 2020.

[21] M. A. Tawhid and A. M. Ibrahim, "A hybridization of grey wolf optimizer and differential evolution for solving nonlinear systems", *Evol. Syst.*, vol. 11, no. 1, pp. 65–87, 2020.

[22] M.-T. Vakil-Baghmisheh and M. Salim, "The design of PID controllers for a Gryphon robot using four evolutionary algorithms: a comparative study", *Artif. Intell. Rev.*, vol. 34, no. 2, pp. 121–132, 2010.

[23] D. L. Gonzalez-Alvarez, M. A. Vega-Rdriguez and A. Rubio-Largo, "Searching for common patterns of protein sequences by means of a parallel hybrid honey-bee mating optimization," *Parallel Computing*, vol. 76, pp. 1-17, 2018.

[24] A. S. Al-Araji, "An adaptive swing-up sliding mode controller design for a real inverted pendulim system based on Culture-Bees algorithm," *European Journal of Control*, vol. 45, pp. 45-56, 2018.

[25] A. Baniamerian, M. Bashiri and R. Tavakkoli-Moghaddam, "Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking," *Applied Soft Computing Journal*, vol. 75, pp. 441-460, 2018.

[26] S. C. and A. T., "Internet of medical things-load optimization of power flow based on hybrid enhanced grey wolf optimization and dragonfly algorithm," *Future Generatin Computer Systems*, vol. 98, pp. 319-330, 2019.

[27] H. Zhang, Q. Zhang, L. Ma, Z. Zhang and Y. Liu, "A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows", *Information Science*, vol. 490, pp. 166-190, 2019.

[28] A. Rubio-Largo, M. A. Vega-Rodriguez and D. L. Gonzalez-Alvarez, "Hybrid multiobjective artificial bee colony for multiple sequence alignment", *Applied Soft Computing*, vol. 41, pp. 157-168, 2015.

[29] A. Rafiee, P. Moradi, A. Ghaderzadeh, "A Swarm Inteligence Based Multi-Lable Feature Selection Method Hybridized with a Local Search Strategy", *Tabriz Journal of Electrical Engineering (TJEE)*, vol. 51, no. 4, Winter 2021

[30] N. Ghorbani, E. Babaei and F. Sadikoglu, "BEMA: Binary Exchange Market Algorithm," *Procedia Computer Science* , vol. 120, pp. 656-663, 2017.

[31] N. Ghorbani, E. Babaei and F. Sadikoglu, "Exchange market algorithm for multi-objective economic emission dispatch and reliability," *Procedia Computer Science*, vol. 120, pp. 633-640, 2017.

[32] T. Dokeroglu, E. Sevinc and A. Cosar, "Artificial bee colony optimization for the quadratic assignment problem," *Applied Soft Computing Journal*, vol. 76, pp. 595-606, 2019.

[33] W. Hu, G. Wen, A. Rahmani and Y. Yu, "Distributed consensus tracking of unknown nonlinear chaotic delayd fractional-order multi-agent systems with external disturbances base on ABC algorithm", *Commun Nonlinear Sci Numer Simulat*, vol. 71, pp. 101-117, 2018.

[34] S. V. Devaraj et al., "Robust Queen Bee Assisted Genetic Algorithm (QBGA) Optimized Fractional Order PID (FOPID) Controller for Not Necessarily Minimum Phase Power Converters", *in IEEE Access*, vol. 9, pp. 93331-93337, 2021, doi: 10.1109/ACCESS.2021.3092215.

[35] M. T. Vakil-Baghmisheh and M. Salim, "A Modified Fast Marriage in Honey Bee Optimization Algorithm," *5th International Symposium on Telecommunications*, pp. 950-955, 2010.

[36] S.Yurish, "Advances in Artificial Intelligence: Reviews", *IFSA Publishing*, 2019.

[37] Q. Duan, S. Sorooshian and V. Gupta, "Effective and Efficient Global Optimization for Conceptual Rainfall-Runoff Models", *Water Esourcesr Esearc*, vol. 28, no. 4, pp. 1015-1031, 1992.

[38] R. Dash, R. Rautray, and R. Dash, "Utility of a Shuffled Differential Evolution algorithm in designing of a Pi-Sigma Neural Network based predictor model", *Appl. Comput. Inform.*, vol. 19, no. 1/2, pp. 22–40, 2023.

[39] V. C. Mariani, L. G. J. Luvizotto, F. A. Guerra and L. dos. S. Coelho, "A hybrid shuffled complex evolution approach based on differential evolution for unconstrained optimization," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5822-5829, 2011.

[40] C. Blum and A. Roli, "Hybrid Metaheuristics: An Introduction", *Studies in Computational Intelligence (SCI)*, vol. 114, pp. 1–30 , 2008.

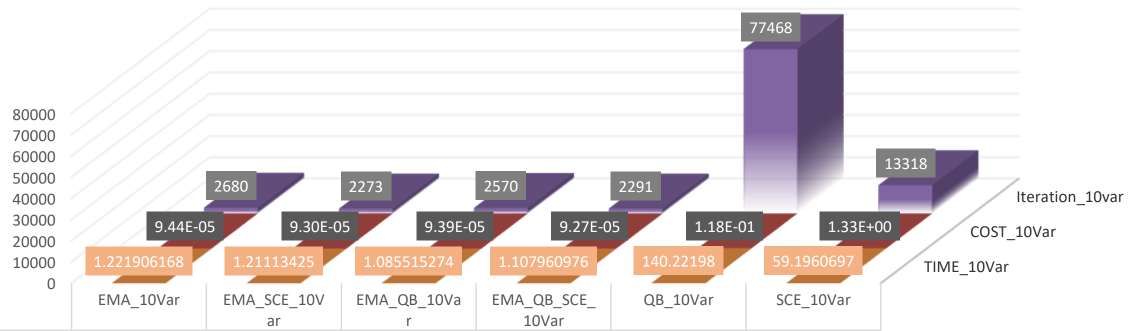
[41] A. F. Ali, M. A. Tawhid, "A hybrid particle swarm optimization and genetic algorithm with population partitioning for large-scale optimization problems", *Ain Shams Engineering Journal*, vol. 8, no. 2, pp. 191-206, 2017.

[42] S. Fidanova, M. Paprzycki and O. Roeva, "Hybrid GA-ACO Algorithm for a model parameters identification problem," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, 2014.

[43] E Mahdipour, M Ghasemzadeh, "A Hybrid Meta-heuristic Algorithm for High Performance Computing", *Tabriz Journal of Electrical Engineering (TJEE)*, vol. 51, no. 1, Spring 2021

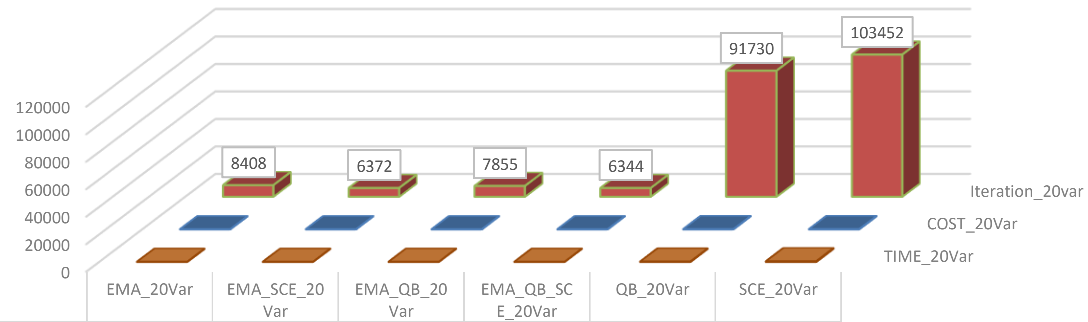
[44] X. Yu and Mitsuo, "Introduction to evolutionary algorithms", *2010th ed. London, England: Springer*, 2012.

[45] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning", *Boston, MA: Addison Wesley*, 1989.



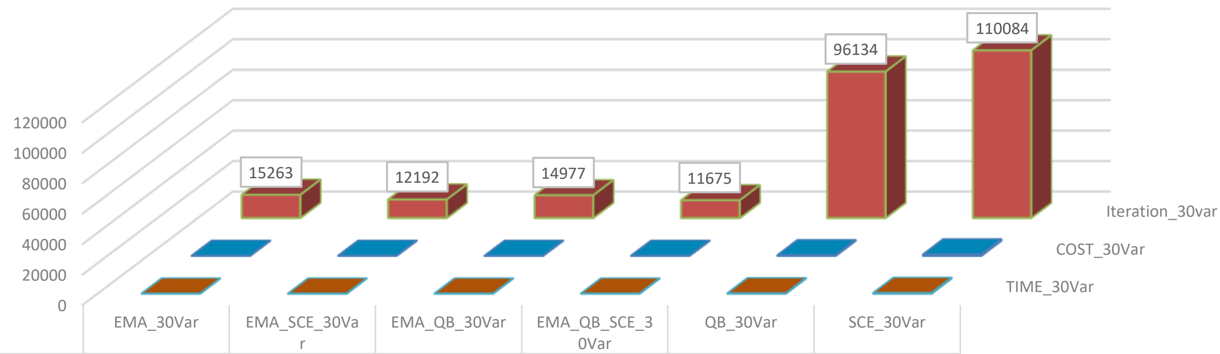
	EMA_10Var	EMA_SCE_10Var	EMA_QB_10Var	EMA_QB_SCE_10Var	QB_10Var	SCE_10Var
TIME_10Var	1.221906168	1.21113425	1.085515274	1.107960976	140.22198	59.1960697
COST_10Var	9.44E-05	9.30E-05	9.39E-05	9.27E-05	1.18E-01	1.33E+00
Iteration_10var	2680	2273	2570	2291	77468	13318

Fig. 4. number of iterations, cost function and time values for 10 variables



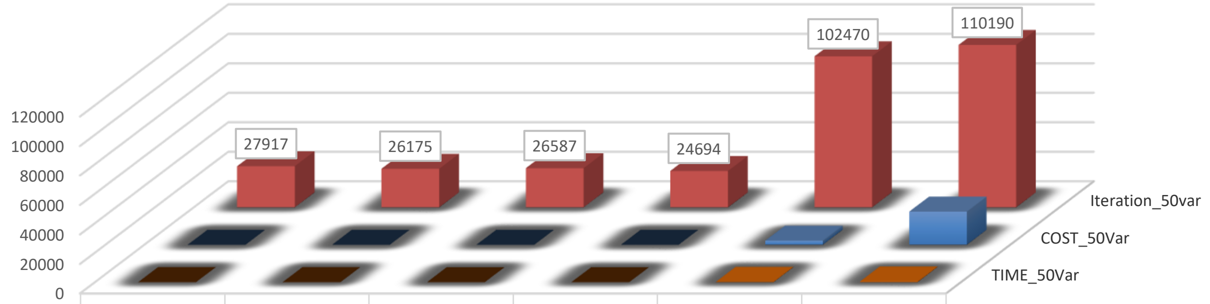
	EMA_20Var	EMA_SCE_20Var	EMA_QB_20Var	EMA_QB_SCE_20Var	QB_20Var	SCE_20Var
TIME_20Var	4.729900751	4.366643927	3.955787931	3.640756355	174.5987157	412.7332424
COST_20Var	1.02E-04	1.00E-04	1.01E-04	1.01E-04	6.15E+01	6.21E+01
Iteration_20var	8408	6372	7855	6344	91730	103452

Fig. 5. number of iterations, cost function and time values for 20 variables



	EMA_30Var	EMA_SCE_30Var	EMA_QB_30Var	EMA_QB_SCE_30Var	QB_30Var	SCE_30Var
TIME_30Var	11.45940158	10.89708922	9.60266302	8.524209954	191.5421849	443.5878371
COST_30Var	3.32E-01	1.04E-04	1.38E-01	1.03E-04	3.76E+02	9.14E+02
Iteration_30var	15263	12192	14977	11675	96134	110084

Fig. 6. number of iterations, cost function and time values for 30 variables



	EMA_50Var	EMA_SCE_50Var	EMA_QB_50Var	EMA_QB_SCE_50Var	QB_50Var	SCE_50Var
TIME_50Var	40.89230814	41.9462746	33.20408418	30.20665955	220.706171	453.5237087
COST_50Var	6.05E+00	2.26E+00	6.34E+00	2.56E+00	2.77E+03	2.25E+04
Iteration_50var	27917	26175	26587	24694	102470	110190

Fig. 7. number of iterations, cost function and time values for 50 variables