Shahid Bahonar University of Kerman
Journal of Mahani Mathematical Research
Print ISSN: 2251-7952 Online ISSN: 2645-4505

EXPECTED BINARY PARTICLE SWARM OPTIMIZATION

M.B. Dowlatshahi ✉ and S. Beiranvand

Article type: Research Article
(Received: 19 January 2025, Received in revised form 15 July 2025)
(Accepted: 09 August 2025, Published Online: 09 August 2025)

Abstract. Binary Particle Swarm Optimization (BPSO), proposed by Kennedy and Eberhart, extends PSO to binary search spaces. However, BPSO suffers from computational complexity due to velocity-based position updates and limited scalability in high-dimensional problems. In this paper, we introduce Expected BPSO (EBPSO), a simplified and faster variant that removes the velocity component and directly uses a probabilistic position update mechanism inspired by expected particle behavior. We theoretically analyze EBPSO's convergence and evaluate its performance across two domains: (1) ten scalable binary benchmark functions (F1–F10) and (2) feature selection for classification using four real-world datasets (Breast Cancer, Iris, Wine, and Digits). EBPSO consistently outperforms BPSO, Binary GA, and other recent binary metaheuristics (e.g., BDO, BSCA, BGWO, BRKO) in both solution quality and runtime. For example, EBPSO achieved up to 15× speedup over BPSO and maintained a competitive advantage across all tested dimensions.

In the feature selection task, EBPSO was used within a wrapper model using an SVM classifier. It reached accuracies of 99.07% on Digits, 99.44% on Wine, and 98.42% on Breast Cancer datasets while selecting fewer features than other methods. Statistical significance was confirmed using paired t-tests and Wilcoxon signed-rank tests, both yielding p-values ¡ 0.01 across all evaluations. Overall, EBPSO demonstrates superior performance, scalability, and statistical robustness, making it a promising tool for large-scale binary optimization and efficient feature selection.

Keywords: Binary Particle Swarm Optimization, Binary optimization problems, Expected Behavior, Convergence Analysis.
2020 MSC: 68T20, 90C59, 68W40

1. Introduction

Particle Swarm Optimization (PSO) is one of the most widely used swarm intelligence algorithms inspired by the social behavior of bird flocking or fish schooling [1], [2]. Since then, various fields have used and evaluated the PSO algorithm. The evaluation results show the positive impact of this algorithm in multiple fields. In [3] researchers have presented an in-depth analysis of the existing PSO-based scheduling schemes in a cloud computing environment and provided a classification of these schemes based on the type of the PSO

✉ dowlatshahi.mb@lu.ac.ir, ORCID: 0000-0002-4862-0626
https://doi.org/10.22103/jmmr.2025.24703.1752
Publisher: Shahid Bahonar University of Kerman

How to cite: M.B. Dowlatshahi, S. Beiranvand, *Expected binary particle swarm optimization*, J. Mahani Math. Res. 2027; 16(1): 1-46.

algorithm which has been used in these solutions. In [4], by presenting an overview of the PSO algorithm, researchers have explained the basic concepts and parameters of PSO, along with a variety of the advancements of the PSO algorithm. From the current literature survey on the PSO algorithm, they have also found that a large amount of research has been carried out on the PSO algorithm, but more areas of the applications of PSO can be enhanced in future research.

Researchers in [5] analyze its present situation of research and application in algorithm structure, parameter selection, topology structure, discrete PSO algorithm and parallel PSO algorithm, multi-objective optimization PSO and its engineering applications. In [6] presents a chronological literature survey on parallel PSO algorithm, its developed variants, implementation platform and strategy, applied area and the objective of addressed problem. Then, all the surveyed articles are further evaluated for concluding the popular parallelization strategy and communication topology. In [7], the particle swarm optimization (PSO) algorithm and a 1D numerical model were utilized to determine temperature-dependent thermal conductivity and specific heat of beech wood and polymethyl methacrylate (PMMA). Using the obtained thermodynamics of wood under constant HF, the numerical model successfully captured the surface temperature at time-dependent HFs. Meanwhile, comparisons using wood temperatures at constant HFs and PMMA temperatures at linear HFs also verified the feasibility of PSO.

A detailed overview of the original PSO and some PSO variant algorithms is presented in [8]. An up-to-date review is provided on the development of PSO variants, which include four types i.e., the adjustment of control parameters, the newly-designed updating strategies, the topological structures, and the hybridization with other optimization algorithms. [9] explores particle swarm optimization (PSO) in the rapidly evolving landscape of biomedical technologies. The authors delve into innovative uses of PSO in various biomedical fields, including image enhancement, data clustering, and drug development, highlighting how PSO contributes to more accurate diagnoses, efficient treatment plans, and streamlined research methodologies. [10] Surveys the published papers in PSO algorithms. Future researchers can use the collected data in this survey as baseline information on the PSO and PSO's applications. In [11], a new path-planning algorithm, NMOPSO, has been introduced to generate Pareto optimal paths for UAVs considering their kinematic constraints. Comparison results show that the NMOPSO outperforms other PSO variants and state-of-the-art metaheuristic optimization algorithms in most criteria including path length, safety and smoothness. In [12], researchers proposed two novel parameter-less niching versions of PSO, called Topologically Nearest-Better Fuzzy PSO (TNB-FPSO) and Distance-based Nearest-Better Fuzzy PSO (DNB-FPSO), for solving multimodal optimization problems. Empirical evaluation of the TNB-FPSO and DNB-FPSO on several complex multimodal benchmark functions demonstrate that the proposed algorithms can obtain better results than the other compared multimodal optimization algorithms in the terms of final solution set quality.

The original version of PSO is merely ideal for solving continuous optimization problems. Hence, the original version of this algorithm is also called Continuous PSO (CPSO). When we're trying to solve a binary optimization problem, using CPSO is not recommended because all variables of the problem are binary and the continuity of variables that is one of the basic assumptions of CPSO does not exist. In other words, the position updating process of CPSO cannot be directly used for binary optimization problems, because the position updating in binary optimization problems means switching between 1 and 0 values. To solve binary optimization problems by PSO, a method should be found for changing particles' positions from 1 to 0 or vice versa. We know that the position updating in CPOS is done using the velocities of particles. Hence, the question here is how the concept of velocity must be employed that allows the PSO to update the positions of particles in a binary space. One of the first ideas is to change the position of a particle with the probability of its velocity. This idea is initially used in the standard Binary version of PSO (BPSO), proposed by Kennedy and Eberhart in [13], in which a sigmoid transfer function transforms the velocity of the particle to the probability of taking the value of 1 for the position of the particle.

From a practical view, stochastic search algorithms must start first with exploration and progressively move towards exploitation as the search continues [14]. In CPSO, the inertia weight plays a very important role in balancing this relation between exploration and exploitation [15]. In such a case, a larger inertia weight encourages the CPSO to explore the search space, whereas a smaller inertia weight forces the CPSO to exploit the search space [16]–[19]. Based on these facts, a linearly decreasing method for inertia weight was proposed in [16] to balance between exploration and exploitation. Regardless of the differences between the CPSO and standard BPSO, most studies wrongly used the commonly decreasing inertia weight scheme to design an adaptive inertia weight in standard BPSO. Ref [20] investigated the importance and the effect of the inertia weight parameter on the trade-off between exploration and exploitation of standard BPSO, from both theoretical and empirical perspectives. In short, their research findings suggest that in standard BPSO a smaller inertia weight encourages exploration while a larger inertia weight enhances exploitation. Thus, unlike CPSO, an increasing inertia weight scheme is necessary to balance between exploration and exploitation in standard BPSO.

Even with the correct setting of inertia weight parameter to balance between exploration and exploitation, the standard BPSO suffers from two major issues due to mere imitation of the original PSO structure: 1) complex understanding of the position generation, and 2) the low efficiency of the algorithm especially in solving the large-scale binary optimization problems. By analyzing the results found by Ref [20] on the role of the inertia weight parameter in the balance between exploration and exploitation, we conclude that the behavior of standard BPSO in the expected case has two important properties: 1) the current position of a particle does not have any effect on the generation of its next position, and 2) the combination of inertia weight parameter, the velocity of the particles, and the sigmoid transfer function behaves as the probability of convergence of standard BPSO. Hence, we believe that the current position updating scheme in standard BPSO is neither simple nor efficient, and the position updating can be carried out in another simple and efficient manner. This fact leads us to propose a new binary version of PSO that imitates the expected behavior of standard BPSO. The proposed algorithm is called Expected BPSO (EBPSO) and partially resolves the major weaknesses of the standard BPSO. The theoretical convergence analysis of EBPSO to the global optimum solution is carried out, and to evaluate the effectiveness and efficiency of it, several experiments are done on the four scalable benchmark optimization functions.

Although several variants of BPSO have been proposed in prior literature including parameter tuning, adaptive inertia weights, and hybrid models, the majority of these approaches preserve the original velocity-based framework of PSO. This design leads to inherent complexity in position updates and increased computational overhead, particularly in high-dimensional binary spaces. In addition, many recent studies focus on improving accuracy but neglect the interpretability and simplicity of the underlying mechanism. To our knowledge, few studies explicitly aim to simplify the BPSO structure itself while maintaining its exploratory power. This gap motivated us to develop a new variant that eliminates the velocity component and models the position update based on the expected behavior of the BPSO.

Motivated by the complexity and inefficiencies of the original BPSO, we propose a simplified version, Expected BPSO (EBPSO) that directly models the expected particle behavior using a probabilistic rule instead of velocity-based transformations. This approach draws inspiration from compact optimization methods, such as the Compact Genetic Algorithm (cGA)[21], which replace explicit populations with probabilistic models. Similarly, EBPSO eliminates velocity updates and operates over a lightweight probabilistic representation of swarm dynamics, resulting in a more interpretable and efficient binary optimization method, particularly for high-dimensional problems. The main contributions of this paper are summarized as follows:

1) Based on the expected behavior of the standard BPSO in generating the next position of a particle, it is intuitively shown that the standard BPSO suffers from two major issues due to mere imitation of the original PSO structure: complex understanding of the principles of position generation and the low efficiency of the algorithm especially in solving large-scale binary optimization problems.

2) A simple and efficient binary optimization algorithm, EBPSO, is proposed inspired by the expected behavior of standard BPSO.

3) The theoretical convergence analysis of EBPSO is carried out, in which a theorem is derived to prove that the EBPSO converges in probability to the global optimum solution.

4) To evaluate the effectiveness and efficiency of EBPSO, a comparison with standard BPSO is done on the four scalable benchmark optimization functions, and the results are presented. To demonstrate the effectiveness and scalability of the proposed EBPSO, we conduct extensive experiments comparing its performance against ten state-of-the-art binary optimization algorithms on ten benchmark functions and four real-world feature selection tasks.

The structure of this paper is organized as follows. In Section 2, the standard BPSO and its theoretical results regarding the role of inertia weight are reviewed. Section 3 introduces the proposed EBPSO. Section 4 presents the theoretical global convergence analysis. Section 5 reports the experimental results and comparative performance analysis. Finally, Section 6 concludes the paper and outlines future directions.

2. The standard BPSO

In 1995, Kennedy and Eberhart [1] proposed the Continuous Particle Swarm Optimization (CPSO) algorithm for solving continuous optimization problems by simulation a swarm of particles that are flying around the virtual search space. At the beginning of CPSO, a swarm of particles with random positions is generated. Then the position of all particles is iteratively changed in the search space of the problem until the algorithm finds a good solution[22], [23]. The position update rule in CPSO uses both the velocity and current position vectors of a particle to determine the next position of that particle. Let us show each particle as a triple $Particle_i = \left(\vec{X}_i(t), \vec{V}_i(t), P\vec{best}_i(t)\right)$ in an n-dimensional search space where $\vec{X}_i(t)$ and $\vec{V}_i(t)$ are the position and velocity vectors of the i-th particle, respectively, and $P\vec{best}_i(t)$ is the vector of the personal best position found by the particle as follows:

(1) $\vec{X}_i(t) = \left(x_i^1(t), x_i^2(t), \ldots, x_i^d(t), \ldots, x_i^n(t)\right)$, for $i = 1, 2, \ldots, N$.

(2) $\vec{V}_i(t) = \left(v_i^1(t), v_i^2(t), \ldots, v_i^d(t), \ldots, v_i^n(t)\right)$, for $i = 1, 2, \ldots, N$.

(3) $P\vec{best}_i(t) = \left(pbest_i^1(t), pbest_i^2(t), \ldots, pbest_i^d(t), \ldots, pbest_i^n(t)\right)$, for $i = 1, 2, \ldots, N$.

where N is the swarm size or number of particles. Also, suppose we show the best position of the swarm by vector $G\vec{best}(t))$ as follows:

(4) $\vec{Gbest}(t) = \left(gbest^1(t), gbest^2(t), \ldots, gbest^d(t), \ldots, gbest^n(t)\right)$, for $i = 1, 2, \ldots, N$.

The next velocity of the i-th particle is calculated as follows:

(5) $v_i^d(t+1) = w(t) \times v_i^d(t) + c_1 \times r_1 \times \left(pbest_i^d(t) - x_i^d(t)\right) + c_2 \times r_2 \times \left(gbest^d(t) - x_i^d(t)\right)$,

where w(t) is inertia weight, $c1 > 0$ and $c2 > 0$ are acceleration coefficients, r1, and r2 are random numbers in the interval [0, 1], and $v_i^d\,(t+1)$ and $v_i^d\,(t)$ are the next and current velocity of the $i-th$ particle, respectively. Note that most studies limit the value of $v_i^d(t+1)$ into a bound $(|v_i^d\,(t+1)| \leq v_{max})$ regarding the search space bound of the optimization problem that must be solved. In CPSO, the inertia weight $w(t)$ plays a very important role in balancing this relation between exploration and exploitation, so that a larger value of this parameter encourages exploration, while a smaller value of it provides exploitation [6,7,8,9].

To update the next position of $i-th$ particle, the current position of this particle, i.e.$x_i^d(t)$, and the next velocity of it, $v_i^d\,(t+1)$, is used as follows:

(6) $x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$.

The CPSO algorithm is just ideal for solving continuous optimization problems. To solve binary optimization problems, a Binary Particle Swarm Optimization (BPSO) was proposed by Kennedy and Eberhart [13][24]. BPSO and CPSO are distinguished by two different components: 1) a transfer function to map particles' velocities to probability values in the interval [0,1], and 2) a special position updating method by which position vectors could be updated with the probability of their velocities. Therefore, for the $d-th$ bit of the $i-th$ particle, the velocity $v_i^d(t+1)$ is transformed into a probability $s(v_i^d(t+1))$ of taking the value of 1 by the following sigmoid transfer function:

(7) $s\left(v_i^d(t+1)\right) = \frac{1}{1+e^{-v_i^d(t+1)}}$.

Then, based on the value of $s(v_i^d(t+1))$, the bit value $x_i^d(t+1)$ is updated as follows:

(8) $x_i^d(t+1) = \begin{cases} 1, & \text{if } rand < s\left(v_i^d(t+1)\right), \\ 0, & \text{otherwise.} \end{cases}$

where $rand$ is a uniformly distributed random number in the interval [0, 1]. The pseudo-code of the standard framework of BPSO is described in Algorithm 1.

Recently, the effect and role of the inertia weight parameter on the balancing between exploration and exploitation of standard BPSO has been investigated by Ref [20]. Like most theoretical analyses provided for CPSO, theoretical analyses of standard BPSO are also based on the stagnation assumption, in which the personal and global best positions were assumed to remain the same throughout the process. Since the velocity and position updating are carried out "independently" in standard BPSO, the particle index i and the bit index

Algorithm 1. Outline of standard BPSO for maximization.

Initialize c_1, c_2, w, N, and stopping criterion;
$t = 0$;
For $i = 1$ to N do
  Randomly generate the initial solution $\vec{X}_i(t)$;
  $\vec{Pbest}_i(t) = \vec{X}_i(t)$;
  Randomly generate the initial velocity $\vec{V}_i(t)$;
End for
$k = \arg\max_{i=1,\ldots,N} fitness(\vec{Pbest}_i(t))$;
$\vec{Gbest}(t) = \vec{Pbest}_k$;
While stopping criterion is not satisfied do
  Update the value of $w(t)$;
  For $i = 1$ to N do
    For $d = 1$ to n do
      Generate two random numbers r_1 and r_2 in [0,1];
      $v_i^d(t+1) = w(t) \cdot v_i^d(t) + c_1 \cdot r_1 \cdot (pbest_i^d(t) - x_i^d(t))$
        $+c_2 \cdot r_2 \cdot (gbest^d(t) - x_i^d(t))$;
      Generate a random number rand in [0,1];
      If $rand < \dfrac{1}{1+e^{-v_i^d(t+1)}}$ then $x_i^d(t+1) = 1$;
      Else $x_i^d(t+1) = 0$;
    End for
    If $fitness(\vec{X}_i(t+1)) > fitness(\vec{Pbest}_i(t))$
      $\vec{Pbest}_i(t+1) = \vec{X}_i(t+1)$;
    Else $\vec{Pbest}_i(t+1) = \vec{Pbest}_i(t)$;
  End for
  $k = \arg\max_{i=1,\ldots,N} fitness(\vec{Pbest}_i(t+1))$;
  $\vec{Gbest}(t+1) = \vec{Pbest}_k$;
  $t = t + 1$;
End while
Output: $\vec{Gbest}(t)$.

d in Eq. (5) can be safely removed as follows:
(1)
$$v\,(t+1) = w\,(t) \times v\,(t) + c_1 \times r_1 \times (pbest(t) - x(t)) + c_2 \times r_2 \times (gbest(t) - x(t))\,.$$

where $v(t)$ and $x(t)$ are the velocity and the value of the considered bit in the $t-th$ iteration. In standard BPSO, it is assumed that $c1$ and $c2$ are set to constants. Ref [20] has analyzed the behavior of standard BPSO under different $w(t)$, $pbest(t)$, and $gbest(t)$ values. Theorems 2.1 and 2.2 summarize the expected value of $x(t+1)$ when $w(t) = 1$ and $0 < w(t) < 1$, respectively.

Theorem 2.1. *If $w(t) = 1$, then under the standard parameter settings where $c_1 = c_2$ and $v(0) = 0$,*

(1) *If $pbest(t) = gbest(t)$, then $\Pr(x(t+1) = pbest(t))$ is a non-decreasing function of generation t and converges to 1 in expectation.*
(2) *If $pbest(t) \neq gbest(t)$, then $\Pr(x(t+1) = 1)$ and $\Pr(x(t+1) = 0)$ fluctuate around 0.5.*

Theorem 2.2. (1) *If $pbest(t) = gbest(t)$, then*

$$\Pr\left(x(t+1) = pbest(t)\right) \ (or\ \Pr\left(x(t+1) = gbest(t)\right)) > 0.5.$$

(2) *If $pbest(t) \neq gbest(t)$, then the sequences*

$$\Pr\left(x(t+1) = 1\right) \quad and \quad \Pr\left(x(t+1) = 0\right)$$

are expected to fluctuate around 0.5.

Proof. See [20] for proof. □

Remark 2.3. Theorems 2.1 and 2.2 show that a smaller inertia weight encourages standard BPSO to explore, while a larger inertia weight directs standard BPSO to exploit.

3. The proposed Expected BPSO (EBPSO)

Applications and optimized versions of BPSO have been presented in various fields so far [25]–[38]. Inspired form the results found by Ref [20] about the role of the inertia weight parameter in standard BPSO, we propose an Expected version of BPSO (EBPSO) in this Section. For this purpose, we first present some implicit results from the expected behavior of standard BPSO, and then introduce EBPSO for solving binary optimization problems. Taking a look at Theorems 2.1 and 2.2, the following results can be concluded.

- In the expected case, the combination of parameter $0 < w(t) \leq 1$, velocity of particles, and the sigmoid transfer function behaves as the convergence probability of standard BPSO. As we know, the position update rule of standard BPSO uses the velocity vector to determine the movement of the particle in the search space. According to Theorems 2.1 and 2.2, under the standard parameter settings, the expected value of

$$\Pr\big(x(t+1) = pbest(t) = gbest(t)\big)$$

is greater than 0.5 when $0 < w(t) < 1$, and is 1 when $w(t) = 1$. Ref. [20] has shown that given the same iteration t, a smaller $w(t)$ leads standard BPSO to a narrower range of velocity which is closer to 0. This means that in standard BPSO a smaller $w(t)$ delays the convergence while a larger $w(t)$ encourages the convergence.

Based on this observation, we can conclude that the final purpose of the combination of $w(t)$, particles' velocities, and the sigmoid transfer function is to determine the convergence probability of standard BPSO.

According to the overhead of Eqs. (5) and (7), we believe that this combination is neither simple nor efficient for convergence probability calculating, because their role can be played by a simple and efficient convergence probability parameter. By doing so, both calculation and storage of the particles' velocities are useless and can be removed from the algorithm. Also, there is no need to use the sigmoid transfer function to convert the particles' velocities to the probability of taking the value of 1. In this case, the frugality of the memory is of order $N \times n$ and the frugality of the time is of order $N \times n \times T$, where T is the maximal number of iterations. Note that removing the velocity from the algorithm has at least another advantage: we do not need to tune the values of three parameters c_1, c_2, and $v_{\max}$.

- In the expected case, the current position of each particle does not have any effect on the formation of the next position. As we know, the current position of a particle in standard BPSO is only used for calculating the velocity of the particle by Eq. (5). Considering Theorems 2.1 and 2.2, the expected value of $x(t+1)$ under the standard parameter settings is solely generated using the values of three parameters of Eq. (5), i.e., $pbest(t)$, $gbest(t)$, and $w(t)$. This means that we do not need to store the current position of the particles. And since memory is reusable, we only need an $X_{1\times n}$ vector, i.e., $X = (x^1, x^2, \dots, x^d, \dots, x^n)$, to generate all particles. In this case, the frugality of the memory is of order $(N-1) \times n$.

Using the above observations about the expected behavior of standard BPSO (see Theorems 2.1 and 2.2), we can propose the following rules to generate $x(t+1)$ in EBPSO:

$$\Pr\big(x(t+1)\big) = \begin{cases} \Pr\big(x(t+1)=1\big) = \Pr\big(x(t+1)=0\big) = 0.5, \\ \qquad\qquad \text{if } pbest(t) \neq gbest(t), \\ \Pr\big(x(t+1) = pbest(t) = gbest(t)\big) = P(t), \\ \qquad\qquad\qquad\qquad \text{if } pbest(t) = gbest(t), \end{cases}$$

where $P(t)$ is called the *convergence probability*, and we consider $0.5 < P(t) \leq 1$ based on Theorems 2.1 and 2.2. If the value of $P(t)$ is near to 0.5, then the algorithm behaves similar to a "pure random search" algorithm and therefore strongly encourages exploration. For example, if $P(t) = 0.5$, then, regardless of the values of $pbest(t)$ and $gbest(t)$, we always have: $\Pr(x(t+1)=1) = \Pr(x(t+1)=0) = 0.5$. On the other hand, if the value of $P(t)$ is near to 1, then the algorithm strongly promotes exploitation. According to the above descriptions, the bit value $x_i^d(t+1)$ is generated based on the values of $pbest_i^d(t)$,

$gbest^d(t)$, and $P(t)$ as follows:

$$x_i^d(t+1) = \begin{cases} gbest^d(t)\,, & \text{if } pbest_i^d(t) = gbest^d(t) \text{ and } \texttt{rand} < P(t), \\ \sim gbest^d(t)\,, & \text{if } pbest_i^d(t) = gbest^d(t) \text{ and } \texttt{rand} \geq P(t), \\ 0, & \text{if } pbest_i^d(t) \neq gbest^d(t) \text{ and } \texttt{rand} < 0.5, \\ 1, & \text{if } pbest_i^d(t) \neq gbest^d(t) \text{ and } \texttt{rand} \geq 0.5. \end{cases} \tag{10}$$

where `rand` is a uniformly distributed random number in the interval [0, 1]. In practice, we use a time-dependent convergence probability $P(t)$ to control the transition from exploration to exploitation. At the beginning of the search, $P(t)$ is initialized close to 0.5, allowing the algorithm to explore the search space with more randomness. As iterations progress, $P(t)$ gradually increases towards 1.0, emphasizing exploitation around the best solutions found. This dynamic behavior mimics cooling schedules in simulated annealing or inertia decay in standard PSO. The specific schedule for $P(t)$ was selected based on preliminary experiments, balancing convergence speed and solution quality. Empirically, this adaptive control helped EBPSO outperform BPSO and Binary GA in both accuracy and efficiency across all benchmark functions.

The pseudo code of the EBPSO is described in Algorithm 2. Because the principles of position generation in EBPSO are very simple, in addition to efficiency we believe that the understandability of this algorithm is more than standard BPSO. According to Ref. [39], most of end-users of an algorithm are often more interested in understandability than accuracy. In fact, if the end-user understands the main principles of the algorithm, he can assess whether it can serve the task at hand.

Note that in the standard BPSO, not only the interpretation of the velocity and particle trajectories are changed, but also the meaning and behavior of the velocity are reversed. In other words, the behavior of velocity in standard BPSO is in opposition to that in CPSO. Let $v_i^d(t+1)$ be the velocity of particle i in dimension d at iteration $t+1$. In the expected case of standard BPSO, the meaning of $v_i^d(t+1) = 0$ is providing a high probability of changing the particle position in the search space, because we have $\Pr\big(x(t+1) = 1\big) = \Pr\big(x(t+1) = 0\big) = 0.5$. On the other hand, the meaning of a large absolute value for $v_i^d(t+1)$ is providing a small probability of changing the particle position in the search space. Note that this behavior is in contradiction with the concept of velocity in the end-users' mind.

In the last two decades, lack of enough attention to this opposition led some studies to wrongly propose the commonly used decreasing inertia weight scheme in CPSO for the standard BPSO. In our opinion, just a part of this misuse is the mistake of end-users and another part is the weakness of understandability of standard BPSO, which misleads end-users. On the contrary, the meaning of convergence probability $P(t)$ in EBPSO leads the end-users to correctly solve their binary optimization problems: a smaller $P(t)$ encourages exploration, while a larger $P(t)$ enhances exploitation.

Algorithm 2. Outline of EBPSO for maximization.

Initialize $P_{\min}$, $P_{\max}$, N, and stopping criterion;
$t = 0$;
For $i = 1$ to N do
  Randomly generate the initial solution $\vec{X}(t)$;
  $\vec{Pbest}_i(t) = \vec{X}(t)$;
End for
$k = \arg\max_{i=1,\ldots,N} fitness(\vec{Pbest}_i(t))$;
$\vec{Gbest}(t) = \vec{Pbest}_k$;
While stopping criterion is not satisfied do
  Update the value of $P(t)$;
  For $i = 1$ to N do
    For $d = 1$ to n do
      Generate a random number $rand$ in [0,1];
      Update $x_i^d(t+1)$ as follows (see formula (10)):

$$x_i^d(t+1) = \begin{cases} gbest^d(t), & \text{if } pbest_i^d(t) = gbest^d(t) \text{ and } rand < P(t), \\ gbest^d(t), & \text{if } pbest_i^d(t) = gbest^d(t) \text{ and } rand \geq P(t), \\ 0, & \text{if } pbest_i^d(t) \neq gbest^d(t) \text{ and } rand < 0.5, \\ 1, & \text{if } pbest_i^d(t) \neq gbest^d(t) \text{ and } rand \geq 0.5. \end{cases}$$

    End for
    If $fitness(\vec{X}(t+1)) > fitness(\vec{Pbest}_i(t))$ then
      $\vec{Pbest}_i(t+1) = \vec{X}(t+1)$;
    Else
      $\vec{Pbest}_i(t+1) = \vec{Pbest}_i(t)$;
  End for
  $k = \arg\max_{i=1,\ldots,N} fitness(\vec{Pbest}_i(t+1))$;
  $\vec{Gbest}(t+1) = \vec{Pbest}_k$;
  $t = t + 1$;
End while
Output: $\vec{Gbest}(t)$.

3.1. **Computational Complexity.** In the standard BPSO, each iteration involves updating a velocity vector of dimension d, applying a sigmoid function to each element, and performing random sampling to determine position updates. This results in a per-particle complexity of $O(d)$ with significant constant overhead.

In contrast, EBPSO eliminates velocity vectors and sigmoid transformations. Instead, it uses a direct probabilistic update mechanism per bit based on matching between pbest and gbest. This not only maintains an $O(d)$ complexity per iteration, but also significantly reduces computation per bit, especially in large-scale problems.

Empirically, EBPSO achieves approximately $2\times$ faster runtimes than BPSO across all benchmark functions and dimensions, confirming its computational efficiency.

4. Theoretical Convergence Analysis of EBPSO

In this section, we present the theoretical convergence analysis of EBPSO based on probability theory. Therefore, we first present the definition of convergence to a global optimum solution, then demonstrate that any feasible solution in the search space can be generated by EBPSO with a positive probability, and finally prove the global convergence of EBPSO.

Denote x^* to be a global optimum solution of the problem, the global convergence of EBPSO can be defined as follows.

Definition 4.1. Let $\left\{P\vec{best}(t),\ t = 1, 2, \ldots\right\}$ be the sequence of personal best positions found by particles in each iteration, where

$$P\vec{best}(t) = \left\{P\vec{best}_1(t), \ldots, P\vec{best}_N(t)\right\}.$$

It is said that EBPSO converges to the global optimum solution x^*, if and only if

$$\lim_{t\to\infty} \Pr\left\{x^* \in P\vec{best}(t)\right\} = 1.$$

Lemma 4.2. *For $0 < P(t) < 1$, EBPSO can generate any feasible solution in each iteration with a probability greater than zero.*

Proof. Without loss of generality, we consider the process of generating $x_i^d(t+1)$, which is the d-th bit of particle i generated at iteration $t+1$. Denote $pbest_i^d(t)$ to be the d-th personal best bit found by particle i at iteration t, and $gbest^d(t)$ to be the d-th best bit of the entire population at iteration t. Based on Algorithm 2, there are three different cases to be investigated:

(1) If $pbest_i^d(t) \neq gbest^d(t)$, then

$$\Pr\{x_i^d(t+1) = 0\} = \Pr\{x_i^d(t+1) = 1\} = 0.5.$$

(2) If $pbest_i^d(t) = gbest^d(t) = 1$, then

$$\Pr\{x_i^d(t+1) = 1\} = P(t) > 0, \qquad \Pr\{x_i^d(t+1) = 0\} = 1 - P(t) > 0.$$

(3) If $pbest_i^d(t) = gbest^d(t) = 0$, then

$$\Pr\{x_i^d(t+1) = 0\} = P(t) > 0, \qquad \Pr\{x_i^d(t+1) = 1\} = 1 - P(t) > 0.$$

Because the bit $x_i^d(t+1)$ is independently generated in EBPSO, the above cases are satisfied for each particle i and each dimension d. In conclusion, in each iteration EBPSO can generate any feasible solution of the search space $S = \{0,1\}^n$ with a probability greater than zero. □

Theorem 4.3. *For $0 < P(t) < 1$, EBPSO converges in probability to the global optimum solution x^*.*

Proof. Lemma 4.2 shows that there exists a probability $p > 0$ for generating any feasible solution of the search space $S = \{0,1\}^n$ in each iteration. Because the global optimum solution x^* itself is a feasible solution in $S = \{0,1\}^n$, we know that there exists a probability $p > 0$ for generating it. Thus, there exists a probability $q = 1 - p < 1$ for not generating x^* in each iteration. So:

$$\lim_{t\to\infty} \Pr\left\{x^* \in \vec{Pbest}(t)\right\} = 1 - \lim_{t\to\infty} \Pr\left\{x^* \notin \vec{Pbest}(t)\right\} = 1 - \lim_{t\to\infty} q^t = 1.$$

□

5. Experimental Results

In this section, we evaluate the efficiency and effectiveness of the EBPSO on four scalable benchmark optimization functions. These benchmark functions have been commonly used as test problems for binary optimization algorithms in previous works [40–43].

First, we describe the characteristics of the selected benchmarks. Then, we present and analyze the results obtained from the proposed EBPSO and the standard BPSO on these benchmarks. Finally, to examine the effect of different values of the convergence probability parameter on the performance of EBPSO, an experiment has been conducted using various settings of this parameter.

5.1. **Benchmarks Description.** Table 1 presents ten scalable binary optimization benchmark functions designed to evaluate large-scale optimization algorithms. These benchmarks include both unimodal functions (F_1, F_2, F_3, F_5, F_7, and F_9) for testing exploitation capability, and multimodal functions (F_4, F_6, F_8, and F_{10}) for assessing exploration performance.

Each function features unique characteristics such as block-wise structures, linear weighting, deceptive traps, and Gaussian noise. The dimensionality is consistently set across $n = 160, 320, 640$, and 1280 to enable comprehensive scalability analysis.

The benchmark suite provides a systematic framework for evaluating algorithm performance across multiple aspects: exploitation (via unimodal functions), exploration (through multimodal landscapes), noise tolerance (F_7), and structural problem-solving (F_2 and F_8). This carefully designed collection allows for rigorous comparison of binary optimization algorithms under diverse conditions, while maintaining uniform scaling across all test functions for reliable performance assessment. The inclusion of both basic and complex function types ensures balanced evaluation of algorithmic capabilities in solving large-scale binary optimization problems.

TABLE 1. Main characteristics of the binary maximization benchmark functions.

Benchmark function	Range	n	fmax
$F_1(X) = \sum_{i=1}^{n} x_i$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280
$F_2(X) = \sum_{i=1}^{\frac{n}{8}} \left(\prod_{j=8(i-1)+1}^{8i} x_j\right)$	$\{0,1\}$	160	20
		320	40
		640	80
		1280	160
$F_3(X) = \sum_{i=1}^{\lceil \frac{n}{2} \rceil} x_{2i-1} - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} x_{2i}$	$\{0,1\}$	160	80
		320	160
		640	320
		1280	640
$F_4(X) = \left\lvert \frac{n}{2} - \sum_{i=1}^{n} x_i \right\rvert + \frac{n}{2}$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280
$F_5(X) = \max\left(0, \sum_{i=1}^{n} x_i - \frac{n}{2}\right)$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280
$F_6(X) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n+1}$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280
$F_7(X) = \sum_{i=1}^{n} x_i + \mathcal{N}(0, 0.05)$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280
$F_8(X) = \sum_{\text{block}=1}^{\lfloor n/8 \rfloor} f(\text{block})$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280
$F_9(X) = \sum_{i=1}^{n} x_i \cdot \left(1 - \left\lvert \frac{2i}{n} - 1 \right\rvert\right)$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280
$F_{10}(X) = \sum_{k=1}^{\lfloor n/5 \rfloor} f_{\text{trap}}\left(x^{(k)}\right)$	$\{0,1\}$	160	160
		320	320
		640	640
		1280	1280

5.2. **Experimental Setting.** To demonstrate how well the proposed algorithm performs, we applied EBPSO to the above-mentioned benchmark functions and compared the results with those of standard BPSO. To balance exploration and exploitation in standard BPSO, we used the following non-decreasing function suggested in Ref. [20] to update $w(t)$:

$$w(t) = \begin{cases} w_{\min} + \frac{t}{\rho \times T} \times (w_{\max} - w_{\min}), & \text{if } t \leq \rho \times T \\ w_{\max}, & \text{if } \rho \times T < t \leq T \end{cases} \tag{11}$$

where t and T stand for the number of iterations elapsed and the maximal number of iterations, respectively. $w_{\min}$ and $w_{\max}$ are the lower and upper bounds of $w(t)$. The parameter ρ with $0 \leq \rho \leq 1$ controls the number of iterations over which $w(t)$ increases from $w_{\min}$ to $w_{\max}$. To balance exploration and exploitation in EBPSO, we recommend that $P(t)$ be a non-decreasing function of the generation t. In this paper, we update $P(t)$ using the following equation:

$$P(t) = \begin{cases} P_{\min} + \frac{t}{\rho' \times T} \times (P_{\max} - P_{\min}), & \text{if } t \leq \rho' \times T \\ P_{\max}, & \text{if } \rho' \times T < t \leq T \end{cases} \tag{12}$$

where $P_{\min}$ and $P_{\max}$ are the lower and upper bounds of $P(t)$, and $0 \leq \rho' \leq 1$ is the parameter to control the number of iterations to make $P(t)$ increase from $P_{\min}$ to $P_{\max}$. Although such setting is only a special case for updating $P(t)$, experiments show that it helps EBPSO to have satisfactory performance in solving different types of binary optimization problems. More sophisticated configuration approaches for updating $P(t)$ (e.g., self-adaption) will be investigated in future studies. In all of the experiments, the swarm size is set to 50 ($N = 50$) and the maximum number of iterations is set to 500 ($T = 500$). As suggested in Ref. [20], we set $w_{\max} = 1$ and $\rho = 0.9$ for standard BPSO. Also, we set $P_{\max} = 1$ and $\rho' = 0.98$ for EBPSO.

To make a fair comparison between standard BPSO and EBPSO, values from $\{0.3, 0.5, 0.7, 0.9\}$ were selected for $w_{\min}$, and values from $\{0.6, 0.7, 0.8, 0.9\}$ were selected for $P_{\min}$. Note that the ranges of $w_{\min}$ and $P_{\min}$ are chosen based on Theorem 2.2, which states that if $0 < w(t) < 1$, then the expected convergence probability of standard BPSO is greater than 0.5.

In other words, for the same iteration t, a smaller value of $w(t)$ within $(0, 1)$ leads the standard BPSO to a convergence probability closer to 0.5, while a larger value of $w(t)$ within $(0, 1)$ results in a convergence probability closer to 1. All of the experiments were implemented in Python and run on a PC with an Intel 2.2 GHz CPU. The results are averaged over 200 independent runs, and the mean of the best solutions, the standard deviation of the best solutions, and the mean computational time of all runs are reported.

Also, to show the efficiency of EBPSO compared with standard BPSO, the speedup is calculated as in Ref. [44]:

$$\text{Speedup} = \frac{\text{Time}_{\text{BPSO}}}{\text{Time}_{\text{EBPSO}}} \tag{13}$$

where $\text{Time}_{\text{BPSO}}$ and $\text{Time}_{\text{EBPSO}}$ are computational times of standard BPSO and EBPSO, respectively, and speedup is the improvement in speed of EBPSO execution rather than standard BPSO execution. Furthermore, besides EBPSO and standard BPSO, competitive algorithms including Binary Genetic Algorithm (BGA), Binary Sine Cosine Algorithm (BSCA), Binary Ring K-Opt Algorithm (BRKO), Binary Differential Operator (BDO), Binary Zonal Optimization Algorithm (BZOA), Binary Grey Wolf Optimizer (BGWO), Binary Harris Hawks Optimization (BHHO), Binary Whale Optimization Algorithm (BWOA), and Binary Weighted Mean Variant (BWMV) were also evaluated.

The parameters of these comparative algorithms were set based on their respective reference studies and optimized for fair comparison. The variable parameters and their selected values for each algorithm are detailed in the corresponding tables. All experiments were conducted under identical computational environments, and performance metrics such as mean best solution, standard deviation, and runtime were consistently reported.

5.3. **Results and Comparisons.** The performance of the proposed Enhanced Binary Particle Swarm Optimization (EBPSO) algorithm was thoroughly evaluated on all benchmark functions F_1 through F_{10}. Detailed results are presented in Tables 2 to 11, each consisting of sub-tables from (a) to (e), comparing EBPSO against various state-of-the-art binary optimization algorithms.

For the initial benchmark functions F_1 to F_4, results shown in Tables 2.a to 5.e indicate that EBPSO consistently achieves superior mean fitness values, especially at higher convergence probabilities $P_{\min} = 0.9$. These tables also demonstrate lower standard deviations compared to standard BPSO and Binary Genetic Algorithm (BGA), reflecting improved stability and reliability of EBPSO's convergence behavior.

For the more challenging functions F_5 through F_7 (Tables 6.a to 8.e), EBPSO maintains competitive performance, with mean values comparable to or better than other advanced algorithms. The speedup factors reported in these tables confirm that EBPSO runs significantly faster than conventional BPSO and often surpasses the runtime efficiency of Binary GA, particularly in low to medium dimensional spaces.

Function F_8, analyzed in Tables 8.a to 8.e, reveals some instances where EBPSO's mean solution quality is slightly lower than a few competing algorithms. Nevertheless, EBPSO still demonstrates competitive performance with generally faster computational times, which is critical for large-scale optimization scenarios.

Function F_{10}, as detailed comprehensively in Tables 11.a through 11.e, shows that EBPSO achieves the best trade-off between solution quality and computational efficiency across all tested dimensions (160 to 1280). EBPSO outperforms BPSO, BGA, BSCA, BRKO, BDO, BGWO, BHHO, BWOA, and BWMV in terms of both mean fitness values and runtime. The speedup values in

Tables 11.b to 11.e particularly emphasize EBPSO's superior scalability and time efficiency. It is important to note from Tables 7 and 11 that while EBPSO's speedup relative to Binary GA decreases as problem dimensionality increases, it consistently maintains significant speed advantages over standard BPSO and other metaheuristics even in large-scale problems.

This notable computational acceleration is attributed to EBPSO's novel update mechanism, which simplifies the traditional velocity and sigmoid transfer computations in standard BPSO to a more efficient, single-step position generation process. This simplification reduces computational overhead without sacrificing solution quality.

In conclusion, the extensive comparative analyses presented in Tables 2 to 11 and their respective sub-tables clearly demonstrate that EBPSO offers robust, reliable, and scalable performance for a wide range of binary optimization benchmarks, making it a strong candidate for future research and applications in binary optimization.

Table 2.a: The obtained results for function F1 by standard BPSO, BGA and EBPSO.

Function (n)	BGA (mut)				BPSO (w_{min})				EBPSO (P_{min})				Speedup (EBPSO & BGA)	Speedup (EBPSO & BPSO)
	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.6	0.7	0.8	0.9		
F1 (160) Mean	101.37	114.315	114.435	110.13	159.84	159.96	160	160	159.95	159.99	160	160	-	-
F1 (160) STD	3.90	4.1393	4.07	3.79	0.41	0.21	0	0	0.22	0.12	0	0	-	-
F1 (160) Time	20.24	18.39	17.07	16.31	1.73	1.74	1.74	1.74	0.73	0.73	0.74	0.74	22.34	2.36
F1 (320) Mean	192.53	209.475	209.835	203.47	301.78	304.96	309.6	317.44	306.8	311.69	317.02	319.89	-	-
F1 (320) STD	5.42	5.96	5.51	5.20	2.89	2.67	2.32	1.32	2.53	2.36	1.51	0.33	-	-
F1 (320) Time	18.34	18.04	21.11	19.59	2.96	2.97	2.97	2.97	1.20	1.20	1.19	1.19	15.16	2.48
F1 (640) Mean	367.435	390.945	388.895	381.61	544	550.78	562.95	590.29	553.52	566.03	583.77	612.5	-	-
F1 (640) STD	7.91	9.627	7.80	7.66	5.73	6.05	5.29	5.07	5.98	4.92	4.60	3.82	-	-
F1 (640) Time	24.06	25.49	23.69	25.44	5.72	5.80	5.79	5.75	2.19	2.20	2.15	2.20	11.02	2.64
F1 (1280) Mean	709.645	741.01	740.075	727.555	974.81	986.1	1009.37	1061.25	989.57	1010.98	1043.78	1103.56	-	-
F1 (1280) STD	9.82	12.76	11.36	10.599	9.69	10.04	9.46	9.83	7.88	8.29	7.82	8.14	-	-
F1 (1280) Time	35.45	38.38	38.68	36.15	11.45	11.50	11.52	11.52	4.15	4.13	4.09	4.03	8.80	2.80

Table 2.b: The obtained results for function F1 by BSCA, BRKO and EBPSO.

Function (n)	BSCA (a)				BRKO (α)				EBPSO (P_{min})				Speedup (EBPSO & BSCA)	Speedup (EBPSO & BRKO)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F1 (160) Mean	101.16	101.26	101.16	100.755	125.805	126.225	126.495	126.71	159.95	159.99	160	160	-	-
F1 (160) STD	2.2055	2.3372	2.1294	1.9039	1.7824	2.0111	1.99	1.8237	0.22	0.12	0	0	-	-
F1 (160) Time	11.54	10.82	11.54	13.22	24.29	24.18	23.94	32.82	0.73	0.73	0.74	0.74	14.82	14.82
F1 (320) Mean	189.925	189.63	190.07	190.085	238.895	239.825	240.77	241.495	306.8	311.69	317.02	319.89	-	-
F1 (320) STD	2.8495	2.7355	2.9774	2.9219	2.4319	2.4747	2.7834	3.1081	2.53	2.36	1.51	0.33	-	-
F1 (320) Time	23.56	20.31	23.86	23.51	29.37	30.09	30.45	31.85	1.20	1.20	1.19	1.19	17.07	17.07
F1 (640) Mean	362.52	362.525	362.895	362.77	461.315	462.18	462.845	464.215	553.52	566.03	583.77	612.5	-	-
F1 (640) STD	4.3023	4.4113	4.5491	4.591	4.0232	3.9468	4.0213	4.28	5.98	4.92	4.6	3.82	-	-
F1 (640) Time	34.58	32.33	32.45	31.97	37.77	36.93	36.44	35.06	2.19	2.20	2.15	2.20	14.87	14.87
F1 (1280) Mean	699.565	700.69	699.8	700.435	896.005	898.5	899.39	901.965	989.57	1010.98	1043.78	1103.56	-	-
F1 (1280) STD	5.752	6.4128	6.0622	6.3329	5.2091	5.6868	5.7357	6.3296	7.88	8.29	7.82	8.14	-	-
F1 (1280) Time	62.35	60.27	59.9	69.3	51.2	51.24	51.16	50.01	4.15	4.13	4.09	4.03	14.86	12.70

Table 2.c: The obtained results for function F1 by BDO, BZOA and EBPSO.

Function (n)	BDO (α)				BZOA (α)				EBPSO (P_{min})				Speedup (EBPSO & BDO)	Speedup (EBPSO & BZOA)
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F1 (160) Mean	127.025	126.96	126.72	125.135	127.21	127.11	126.785	125.025	159.95	159.99	160	160	-	-
F1 (160) STD	1.9038	1.977	2.0812	1.8485	2.1739	1.8379	1.9284	2.0915	0.22	0.12	0	0	-	-
F1 (160) Time	34.53	38.38	34.81	26.76	27.57	27.87	24.65	25.55	0.73	0.73	0.74	0.74	36.66	36.66
F1 (320) Mean	241.35	241.21	240.4	237.415	241.37	241.51	240.335	237.575	306.8	311.69	317.02	319.89	-	-
F1 (320) STD	2.6865	2.8958	2.9682	2.667	2.6913	3.1859	2.7951	2.6085	2.53	2.36	1.51	0.33	-	-
F1 (320) Time	34.43	34.83	27.08	35.12	28.14	25.53	24.63	27.68	1.20	1.20	1.19	1.19	22.76	22.76
F1 (640) Mean	464.715	463.83	463.01	455.985	464.465	464.01	462.19	456.295	553.52	566.03	583.77	612.5	-	-
F1 (640) STD	4.0366	3.9359	4.6238	3.6421	4.0198	4.005	3.8018	4.2718	5.98	4.92	4.6	3.82	-	-
F1 (640) Time	52.06	46.33	53.55	37.25	31.73	32.42	31.63	32.2	2.19	2.20	2.15	2.20	17.33	14.76
F1 (1280) Mean	902.36	901.06	896.83	886.75	901.52	900.655	896.995	886.06	989.57	1010.98	1043.78	1103.56	-	-
F1 (1280) STD	5.6081	5.5504	6.0184	6.0264	6.2289	6.003	5.9376	5.9268	7.88	8.29	7.82	8.14	-	-
F1 (1280) Time	60.59	67.32	80.87	67.1	43.99	44.89	43.09	47.59	4.15	4.13	4.09	4.03	15.03	10.92

Table 2.d: The obtained results for function F1 by BGWO, BHHO and EBPSO.

Function (n)	BGWO (α)				BHHO (α)				EBPSO (P_{min})				Speedup (EBPSO & BGWO)	Speedup (EBPSO & BHHO)
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9		
F1 (160) Mean	119.485	119.495	120.105	119.835	130.21	132.04	132.27	132.1	159.95	159.99	160	160	-	-
F1 (160) STD	2.6115	2.7441	2.7081	2.6092	2.1693	2.5097	2.3616	2.4698	0.22	0.12	0	0	-	-
F1 (160) Time	13.61	12.49	12.66	11.8	16.94	17.17	18.11	17.65	0.73	0.73	0.74	0.74	16.16	16.16
F1 (320) Mean	230.44	230.2	230.96	230.76	250.395	254.985	255.27	254.39	306.8	311.69	317.02	319.89	-	-
F1 (320) STD	3.7863	4.089	3.8947	4.0339	3.6262	4.0773	4.5052	4.6171	2.53	2.36	1.51	0.33	-	-
F1 (320) Time	20.52	19.78	20.1	20.55	19.56	19.08	20.69	20.22	1.2	1.2	1.19	1.19	16.62	16.44
F1 (640) Mean	449.645	449.31	449.55	448.885	486.95	495.91	498.375	497.97	553.52	566.03	583.77	612.5	-	-
F1 (640) STD	5.9126	5.6933	5.3495	5.5571	7.2296	7.6139	7.4246	7.7672	5.98	4.92	4.6	3.82	-	-
F1 (640) Time	36.4	37.7	58.45	36.84	24.78	25.38	26.06	27.76	2.19	2.2	2.15	2.2	16.93	11.53
F1 (1280) Mean	881.045	882.145	881.005	881.11	955.625	974.425	975.55	978.08	989.57	1010.98	1043.78	1103.56	-	-
F1 (1280) STD	7.9349	8.9389	8.2107	7.5351	12.0027	13.1899	15.2534	14.1278	7.88	8.29	7.82	8.14	-	-
F1 (1280) Time	73.02	93.2	76.06	74.9	31.7	33.95	34.19	38.46	4.15	4.13	4.09	4.03	18.12	7.87

Table 2.e: The obtained results for function F1 by BWOA, BWMV and EBPSO.

Function (n)	BWOA (b)				BWMV (α)				EBPSO (P_{min})				Speedup (EBPSO & BWOA)	Speedup (EBPSO & BWMV)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F1 (160) Mean	125.34	128.325	131.77	132.97	127.195	127.185	127.3	127.345	159.95	159.99	160	160	-	-
F1 (160) STD	3.8671	4.5825	5.3951	5.8753	1.9305	1.9107	1.7916	1.8563	0.22	0.12	0	0	-	-
F1 (160) Time	24.94	21.79	20.09	19.13	24.99	22.13	29.35	28.62	0.73	0.73	0.74	0.74	26.21	26.21
F1 (320) Mean	246.32	253.52	260.47	263.8	241.815	241.525	242.04	241.805	306.8	311.69	317.02	319.89	-	-
F1 (320) STD	5.6655	8.9207	10.2864	10.6028	2.9242	2.9222	2.7583	2.7161	2.53	2.36	1.51	0.33	-	-
F1 (320) Time	24.41	29.58	29.55	23.4	26.02	25.56	25.28	27.22	1.2	1.2	1.19	1.19	19.66	19.66
F1 (640) Mean	486.035	504.215	521.355	527.99	464.495	464.95	465.275	465.245	553.52	566.03	583.77	612.5	-	-
F1 (640) STD	8.5548	17.3652	15.877	16.8065	4.0187	4.0605	3.7867	4.1647	5.98	4.92	4.6	3.82	-	-
F1 (640) Time	28.07	26.93	29.16	27.97	37.2	46.2	40.69	37.7	2.19	2.2	2.15	2.2	12.53	12.53
F1 (1280) Mean	962.07	1003.6	1037.055	1050.985	902.63	903.54	903.515	903.445	989.57	1010.98	1043.78	1103.56	-	-
F1 (1280) STD	15.621	32.6728	36.4165	35.7198	6.1476	6.1505	5.3263	6.0147	7.88	8.29	7.82	8.14	-	-
F1 (1280) Time	38.96	42.71	38.86	37.67	57.41	58.31	57.99	51.36	4.15	4.13	4.09	4.03	9.35	9.35

Table 3.a: The obtained results for function F2 by EBPSO, BPSO, and BGA.

Function (n)	Binary GA (mut)				BPSO (w_{min})				EBPSO (P_{min})				Speedup (EBPSO & BGA)	Speedup (EBPSO & BPSO)
	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.6	0.7	0.8	0.9		
F2 (160) Mean	1.645	1.865	2.395	2.25	8.14	8.5	8.97	9.67	8.61	9.2	9.48	9.67	-	-
F2 (160) STD	0.66	0.83	0.94	0.81	1.18	1.19	1.24	1.43	1.34	1.31	1.39	1.6	-	-
F2 (160) Time	270.94	274.8	270.68	272.86	1.92	1.9	1.9	1.88	0.8	0.79	0.81	0.81	342.63	2.37
F2 (320) Mean	2.105	2.635	3.23	3.01	10.61	11.16	12.17	14.04	11.47	12.55	13.26	14.35	-	-
F2 (320) STD	0.71	1.03	1.07	0.86	1.58	1.6	1.52	1.72	1.54	1.8	1.88	1.91	-	-
F2 (320) Time	520.65	594.65	837.03	533.12	3.13	3.1	3.15	3.14	1.28	1.28	1.27	1.27	409.96	2.45
F2 (640) Mean	2.905	3.975	4.755	3.895	13.49	14.31	15.72	19.28	15.31	16.69	18.31	20.7	-	-
F2 (640) STD	0.864	1.31	1.25	0.93	1.83	1.92	2.07	2.18	2.08	2.04	2.2	2.29	-	-
F2 (640) Time	1104.48	1172	1111.67	1599.51	6.16	6.09	6.2	6.17	2.22	2.26	2.25	2.25	497.51	2.74
F2 (1280) Mean	4.155	6.03	6.275	5.135	16.9	18.23	20.47	25.67	20.73	22.67	25.64	29.6	-	-
F2 (1280) STD	1.05	1.66	1.29	0.99	2.01	2.33	2.22	2.65	2.54	2.65	3.11	3.06	-	-
F2 (1280) Time	2203.03	1966.99	2020.49	2028.58	11.53	11.98	11.51	11.61	4.14	4.16	4.12	4.12	477.42	2.82

Table 3.b: The obtained results for function F2 by EBPSO, BSCA, and BRKO.

Function (n)	BSCA (a)				BRKO (α)				EBPSO (P_{min})				Speedup (EBPSO & BSCA)	Speedup (EBPSO & BRKO)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F2 (160) Mean	2.07	2.08	2.095	2.105	4.65	4.595	4.705	4.75	8.61	9.2	9.48	9.67	-	-
F2 (160) STD	0.274	0.3059	0.3098	0.3794	0.6837	0.5753	0.5727	0.6982	1.34	1.31	1.39	1.6	-	-
F2 (160) Time	83.65	84.13	83.85	83.36	98.71	96.68	99.2	97.71	0.8	0.79	0.81	0.81	105.52	105.52
F2 (320) Mean	2.625	2.595	2.6	2.57	6.405	6.445	6.59	6.645	11.47	12.55	13.26	14.35	-	-
F2 (320) STD	0.5517	0.5753	0.5385	0.5339	0.6715	0.7463	0.7155	0.7739	1.54	1.8	1.88	1.91	-	-
F2 (320) Time	182.13	178.12	158.7	158.59	172.96	170.6	171.9	170.07	1.28	1.28	1.27	1.27	124.87	124.87
F2 (640) Mean	3.36	3.375	3.37	3.41	9.315	9.53	9.65	9.685	15.31	16.69	18.31	20.7	-	-
F2 (640) STD	0.52	0.5426	0.5857	0.5403	0.8921	0.9429	0.9734	0.9725	2.08	2.04	2.2	2.29	-	-
F2 (640) Time	315.36	310.71	308.74	311.13	321.08	321.39	323.08	322.93	2.22	2.26	2.25	2.25	139.07	137.22
F2 (1280) Mean	4.54	4.535	4.59	4.545	14.31	14.45	14.52	14.715	20.73	22.67	25.64	29.6	-	-
F2 (1280) STD	0.6545	0.6073	0.6495	0.6387	1.0648	1.1522	1.1746	1.1017	2.54	2.65	3.11	3.06	-	-
F2 (1280) Time	614.59	619.07	614.7	635.25	615.14	631.69	624.43	619.02	4.14	4.16	4.12	4.12	149.17	149.2

Table 3.c: The obtained results for function F2 by EBPSO, BDO, and BZOA.

Function (n)	BDO (α)				BZOA (α)				EBPSO (P_{min})				Speedup (EBPSO & BDO)	Speedup (EBPSO & BZOA)
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F2 (160) Mean	4.825	4.855	4.765	4.41	4.76	4.75	4.635	4.475	8.61	9.2	9.48	9.67	-	-
F2 (160) STD	0.6741	0.7239	0.721	0.5761	0.68	0.669	0.6647	0.5995	1.34	1.31	1.39	1.6	-	-
F2 (160) Time	115.02	112.86	116.46	121.8	97.18	93.97	109.1	94.64	0.8	0.79	0.81	0.81	142.86	123.01
F2 (320) Mean	6.68	6.585	6.45	6.26	6.765	6.595	6.52	6.15	11.47	12.55	13.26	14.35	-	-
F2 (320) STD	0.7922	0.7828	0.669	0.6946	0.7873	0.6935	0.8183	0.691	1.54	1.8	1.88	1.91	-	-
F2 (320) Time	168.66	157.67	160.94	158.99	191.6	168.28	170.83	166.85	1.28	1.28	1.27	1.27	124.15	124.15
F2 (640) Mean	9.725	9.645	9.605	8.845	9.675	9.7	9.485	8.895	15.31	16.69	18.31	20.7	-	-
F2 (640) STD	0.9997	0.905	0.9105	0.8608	0.7806	0.9747	0.9164	0.8741	2.08	2.04	2.2	2.29	-	-
F2 (640) Time	326.45	346.82	414.85	358.45	312.82	313.18	312.76	319.82	2.22	2.26	2.25	2.25	147.05	139.03
F2 (1280) Mean	14.49	14.73	14.23	13.335	14.665	14.585	14.41	13.405	20.73	22.67	25.64	29.6	-	-
F2 (1280) STD	1.1045	1.1167	1.0849	1.1414	1.2778	1.1103	1.3572	1.0104	2.54	2.65	3.11	3.06	-	-
F2 (1280) Time	621.24	684.95	615.33	586.38	601.37	615.53	607.36	599.71	4.14	4.16	4.12	4.12	142.33	142.33

Table 3.d: The obtained results for function F2 by EBPSO, BGWO, and BHHO.

Function (n)	BGWO (a)				BHHO (a)				EBPSO (P_{min})				Speedup (EBPSO & BGWO)	Speedup (EBPSO & BHHO)
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9		
F2 (160) Mean	2.825	2.84	2.87	2.885	5.4	5.655	5.62	5.595	8.61	9.2	9.48	9.67	-	-
F2 (160) STD	0.6741	0.7172	0.7093	0.7757	0.7874	0.7912	0.8807	0.9225	1.34	1.31	1.39	1.6	-	-
F2 (160) Time	90.92	85	94.07	97.44	90.85	96.37	93.78	95.05	0.8	0.79	0.81	0.81	107.59	107.59
F2 (320) Mean	4.34	4.185	4.31	4.225	7.72	8.35	8.41	8.495	11.47	12.55	13.26	14.35	-	-
F2 (320) STD	0.8969	0.819	0.9402	0.9297	0.9064	1.0989	1.0639	1.1314	1.54	1.8	1.88	1.91	-	-
F2 (320) Time	169.46	198.63	214.01	224.38	165.86	168.25	169.08	168.36	1.28	1.28	1.27	1.27	133.43	130.6
F2 (640) Mean	6.75	6.78	6.75	6.675	12.115	13.17	13.225	13.36	15.31	16.69	18.31	20.7	-	-
F2 (640) STD	1.256	1.2048	1.1169	1.086	1.3348	1.4701	1.5081	1.5622	2.08	2.04	2.2	2.29	-	-
F2 (640) Time	332.77	430.88	390.41	391.62	313.16	323.63	375.87	323.9	2.22	2.26	2.25	2.25	149.9	139.18
F2 (1280) Mean	10.905	10.84	10.87	10.845	19.325	21.9	21.925	22.355	20.73	22.67	25.64	29.6	-	-
F2 (1280) STD	1.5349	1.4437	1.4741	1.4321	1.9898	2.2271	2.5825	2.7201	2.54	2.65	3.11	3.06	-	-
F2 (1280) Time	748.81	720.6	982.18	873.49	634.72	616.31	658.46	626.84	4.14	4.16	4.12	4.12	174.9	154.06

Table 3.e: The obtained results for function F2 by EBPSO, BWOA, and BWMV.

Function (n)	BWOA (b)				BWMV (α)				EBPSO (P_{min})				Speedup (EBPSO & BWOA)	Speedup (EBPSO & BWMV)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F2 (160) Mean	3.55	4.125	4.635	5.29	4.805	4.795	4.795	4.75	8.61	9.2	9.48	9.67	-	-
F2 (160) STD	1.0037	1.1702	1.4463	1.583	0.661	0.6269	0.6656	0.6225	1.34	1.31	1.39	1.6	-	-
F2 (160) Time	99.92	134.67	120.01	111.27	113.98	101.44	104.16	102.28	0.8	0.79	0.81	0.81	126.48	140.85
F2 (320) Mean	5.695	6.78	8.28	9.645	6.62	6.695	6.77	6.69	11.47	12.55	13.26	14.35	-	-
F2 (320) STD	1.1798	1.8552	2.3413	2.5882	0.725	0.7629	0.7982	0.7375	1.54	1.8	1.88	1.91	-	-
F2 (320) Time	191.17	170.6	165.85	163.87	184.25	179.37	180.17	180.53	1.28	1.28	1.27	1.27	129.03	129.03
F2 (640) Mean	9.965	12.9	15.93	17.715	9.685	9.79	9.795	9.73	15.31	16.69	18.31	20.7	-	-
F2 (640) STD	1.9656	3.6263	4.054	4.7322	0.8976	0.8694	0.9236	0.8758	2.08	2.04	2.2	2.29	-	-
F2 (640) Time	307.55	305.79	312.79	309.59	339.03	335.51	335.38	353.56	2.22	2.26	2.25	2.25	137.74	135.91
F2 (1280) Mean	18.105	23.61	31.01	35.81	14.62	14.615	14.745	14.715	20.73	22.67	25.64	29.6	-	-
F2 (1280) STD	3.0023	5.8368	8.2934	8.3471	1.16	1.2234	1.109	1.1721	2.54	2.65	3.11	3.06	-	-
F2 (1280) Time	605.49	694.81	640.04	606.76	654.64	682.61	1069.42	922.61	4.14	4.16	4.12	4.12	146.96	147.27

Table 4.a: The obtained results for function F3 by EBPSO, BPSO, and BGA.

Function (n)	BGA (mut)				BPSO (w_{min})				EBPSO (P_{min})				Speedup (EBPSO & BGA)	Speedup (EBPSO & BPSO)
	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.6	0.7	0.8	0.9		
F3 (160) Mean	21.245	34.035	34.595	30.83	79.81	79.93	80	80	79.97	80	80	80	-	-
F3 (160) STD	3.73	4.61	4.15	3.82	0.41	0.26	0	0	0.17	0	0	0	-	-
F3 (160) Time	37.38	36.73	33.42	31.89	1.8	1.8	1.81	1.81	0.92	0.91	0.92	0.9	35.43	1.98
F3 (320) Mean	32.135	49.19	49.075	43.7	141.8	145.29	149.49	157.39	147.26	151.92	156.98	159.88	-	-
F3 (320) STD	5.61	6.01	5.66	5.48	2.9	2.76	2.38	1.48	2.44	2.42	1.63	0.34	-	-
F3 (320) Time	31.32	32.12	30.86	33.02	3.58	3.4	3.68	3.37	1.52	1.52	1.49	1.48	20.85	2.33
F3 (640) Mean	48.455	71.755	68.795	61.625	224.83	231.22	243.29	270.99	233.36	246.12	264.07	292.87	-	-
F3 (640) STD	7.267	8.64	7.78	7.81	5.84	6.37	5.73	5.09	6.19	5.7	5.14	3.74	-	-
F3 (640) Time	35.58	36.89	40.86	56.31	6.28	6.41	6.45	6.37	2.51	2.53	2.52	2.5	14.23	2.54
F3 (1280) Mean	70.565	101.7	98.22	86.59	335.32	347.49	367.67	421.11	349.46	370.15	402.61	463.8	-	-
F3 (1280) STD	10.96	12.44	11.621	10.03	8.86	10.49	9.11	9.1	9.2	9.58	8.11	8.1	-	-
F3 (1280) Time	65.67	55.66	53.89	55.45	12.34	12.36	12.35	12.4	4.7	4.68	4.7	4.65	11.59	2.64

Table 4.b: The obtained results for function F3 by EBPSO, BSCA, and BRKO.

Function (n)	**BSCA** (a)				**BRKO** (α)				**EBPSO** (P_{min})				**Speedup (EBPSO & BSCA)**	**Speedup (EBPSO & BRKO)**
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F3 (160) Mean	20.77	21.305	21.305	21.33	22.625	23.03	23.01	23.57	79.97	80	80	80	-	-
F3 (160) STD	2.0993	2.3879	2.1799	2.1473	1.8666	1.8355	2.1886	2.1107	0.17	0	0	0	-	-
F3 (160) Time	14.88	15.15	15.98	14.92	28.56	27.96	27.31	28.33	0.92	0.91	0.92	0.9	16.53	16.58
F3 (320) Mean	30.12	29.97	30.395	29.685	32.615	32.655	33.01	33.55	147.26	151.92	156.98	159.88	-	-
F3 (320) STD	3.4388	3.1175	3.2216	2.4911	2.716	2.9131	3.0659	3.0623	2.44	2.42	1.63	0.34	-	-
F3 (320) Time	20.83	21.34	22.36	21	31.29	32.4	30.99	32.98	1.52	1.52	1.49	1.48	14.07	14.19
F3 (640) Mean	42.375	42.77	42.23	42.66	46.06	45.78	46.805	47.57	233.36	246.12	264.07	292.87	-	-
F3 (640) STD	4.1188	4.4042	4.643	4.537	4.2856	3.9245	3.9896	4.4794	6.19	5.7	5.14	3.74	-	-
F3 (640) Time	35.03	33.73	35.47	34.37	39.84	40.5	41	39.03	2.51	2.53	2.52	2.5	13.49	13.49
F3 (1280) Mean	59.61	59.645	59.01	61	64.685	65.255	65.665	67.785	349.46	370.15	402.61	463.8	-	-
F3 (1280) STD	6.4256	5.6133	5.2924	6.039	5.9309	5.6036	5.5338	6.1195	9.2	9.58	8.11	8.1	-	-
F3 (1280) Time	64.72	67.01	62.33	61.72	56.69	54.93	55.75	57.2	4.7	4.68	4.7	4.65	13.27	12.19

Table 4.c: The obtained results for function F3 by EBPSO, BZOA, and BDO.

Function (n)	**BDO** (α)				**BZOA** (α)				**EBPSO** (P_{min})				**Speedup (EBPSO & BDO)**	**Speedup (EBPSO & BZOA)**
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F3 (160) Mean	24.06	23.68	23.63	23.72	23.595	23.635	23.81	24.09	14.73	80	80	80	-	-
F3 (160) STD	2.2263	2.0143	2.2278	2.2807	2.1263	2.1099	2.1433	2.3199	3.1316	0	0	0	-	-
F3 (160) Time	25.64	23.82	24.57	24.64	25.6	26.36	26.81	25.62	17.44	0.91	0.92	0.9	26.47	26.47
F3 (320) Mean	33.905	33.8	33.63	33.62	33.895	33.685	33.5	33.89	21.02	151.92	156.98	159.88	-	-
F3 (320) STD	3.0077	2.985	2.8095	3.1041	3.1897	2.9284	2.7713	3.0212	3.8678	2.42	1.63	0.34	-	-
F3 (320) Time	27.16	29.13	27.8	26.57	30.34	29.08	30.24	28.83	28.45	1.52	1.49	1.48	17.95	17.95
F3 (640) Mean	47.6	47.695	47.51	47.145	47.29	47.725	47.925	48.09	30.165	246.12	264.07	292.87	-	-
F3 (640) STD	4.0571	3.9461	4.286	4.0415	4.3642	4.3023	3.8012	4.4723	6.1942	5.7	5.14	3.74	-	-
F3 (640) Time	34.55	33.88	36.08	35.89	36.64	35.52	36.27	35.96	46.21	2.53	2.52	2.5	13.55	13.55
F3 (1280) Mean	67.7	68.285	66.77	67.91	68.365	68.15	67.57	68.32	42.59	370.15	402.61	463.8	-	-
F3 (1280) STD	5.7044	6.3391	5.4997	6.2587	6.0796	6.0735	5.8442	6.4752	7.5942	9.58	8.11	8.1	-	-
F3 (1280) Time	44.43	45.68	45.07	43.88	48.16	48.43	46.78	49.08	73.81	4.68	4.7	4.65	9.44	9.44

Table 4.d: The obtained results for function F3 by EBPSO, BGWO, and BHHO.

Function (n)	**BGWO** (a)				**BHHO** (a)				**EBPSO** (P_{min})				**Speedup (EBPSO & BGWO)**	**Speedup (EBPSO & BHHO)**
	0.5	1	1.5	2	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9		
F3 (160) Mean	14.98	14.975	15.34	24.665	24.225	24.035	23.655	5.595	79.97	80	80	80	-	-
F3 (160) STD	3.0099	3.047	3.0355	2.0549	2.1434	2.2746	2.3078	0.9225	0.17	0	0	0	-	-
F3 (160) Time	17.94	17.72	18.68	20.64	21	21.11	22.11	95.05	0.92	0.91	0.92	0.9	19.69	19.69
F3 (320) Mean	21.545	21.405	20.96	35.195	34.75	34.4	33.62	8.495	147.26	151.92	156.98	159.88	-	-
F3 (320) STD	3.9278	4.5838	4.393	3.1156	3.0931	2.9983	3.1298	1.1314	2.44	2.42	1.63	0.34	-	-
F3 (320) Time	24.92	28.11	25.1	23.02	22.8	25.18	24.98	168.36	1.52	1.52	1.49	1.48	15.55	15.41
F3 (640) Mean	30.915	30.235	29.745	49.805	48.755	48.565	47.59	13.36	233.36	246.12	264.07	292.87	-	-
F3 (640) STD	6.4867	6.3348	6.0067	4.4099	4.0193	4.2797	4.6143	1.5622	6.19	5.7	5.14	3.74	-	-
F3 (640) Time	40.07	42.36	41.69	28.69	28.86	30.36	30.17	323.9	2.51	2.53	2.52	2.5	11.48	11.48
F3 (1280) Mean	41.39	43.325	42.355	69.44	69.285	68.74	68.26	22.355	349.46	370.15	402.61	463.8	-	-
F3 (1280) STD	8.0373	8.1957	8.1049	5.9545	5.833	5.9793	6.6312	2.7201	9.2	9.58	8.11	8.1	-	-
F3 (1280) Time	76.95	74.94	73.12	36.96	37.33	40.75	39.47	626.84	4.7	4.68	4.7	4.65	7.95	7.95

Table 4.e: The obtained results for function F3 by EBPSO, BWOA, and BWMV.

Function (n)	BWOA (b)				BWMV (α)				EBPSO (P_{min})				Speedup (EBPSO & BWOA)	Speedup (EBPSO & BWMV)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F3 (160) Mean	15.54	16.035	15.445	15.795	23.955	23.77	23.905	23.84	79.97	80	80	80	-	-
F3 (160) STD	2.8911	3.0222	3.187	3.0452	2.0477	2.3973	2.0411	1.8424	0.17	0	0	0	-	-
F3 (160) Time	23.6	23.76	24.75	23.86	37.96	28.74	27.53	26.72	0.92	0.91	0.92	0.9	26.22	26.4
F3 (320) Mean	22.37	22.23	21.855	21.555	34.065	34.03	33.78	33.975	147.26	151.92	156.98	159.88	-	-
F3 (320) STD	4.342	4.221	4.1838	4.3055	2.9868	2.9798	3.236	2.8043	2.44	2.42	1.63	0.34	-	-
F3 (320) Time	25.95	31.86	30.24	27.45	31.09	40.6	54.98	59.6	1.52	1.52	1.49	1.48	17.53	18.55
F3 (640) Mean	31.445	31.06	31.035	30.665	47.565	47.92	47.62	48.125	233.36	246.12	264.07	292.87	-	-
F3 (640) STD	6.1088	5.759	5.8859	6.4291	4.097	4.1393	4.0269	4.3577	6.19	5.7	5.14	3.74	-	-
F3 (640) Time	31.55	32.34	31.67	32.36	98.17	77.27	63.41	60.56	2.51	2.53	2.52	2.5	12.62	12.67
F3 (1280) Mean	44.32	44.03	45.12	44.23	67.455	67.885	68.79	68.105	349.46	370.15	402.61	463.8	-	-
F3 (1280) STD	8.09	9.1482	8.1594	8.6924	6.0248	5.3246	6.951	6.5065	9.2	9.58	8.11	8.1	-	-
F3 (1280) Time	39.68	43.03	41.39	41.08	58.15	55.01	53.39	57.51	4.7	4.68	4.7	4.65	8.53	8.83

Table 5.a: The obtained results for function F4 by EBPSO, BPSO, and BGA.

Function (n)	BGA (mut)				BPSO (w_{min})				EBPSO (P_{min})				Speedup (EBPSO & GA)	Speedup (EBPSO & BPSO)
	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.6	0.7	0.8	0.9		
F4(160) Mean	101.38	113.98	113.575	109.825	159.82	159.96	159.99	160	159.94	160	160	160	-	-
F4(160) STD	7	4.49	4.84	3.80	0.4	0.21	0.07	0	0.23	0	0	0	-	-
F4(160) Time	28.54	36.98	25.55	26.15	1.65	1.68	1.69	1.68	0.75	0.75	0.75	0.73	35	2.25
F4(320) Mean	192.45	208.84	207.475	202.67	302.05	304.93	309.63	317.37	307.24	312.01	317.07	319.92	-	-
F4(320) STD	5.20	7.98	7.54	6.03	2.88	2.55	2.21	1.55	2.49	2.21	1.44	0.3	-	-
F4(320) Time	45.68	30.62	28.18	28.39	3.05	3.12	3.1	3.12	1.28	1.28	1.25	1.25	22.54	2.45
F4(640) Mean	368.01	388.685	388.11	379.495	544.77	551.74	562.42	591.15	553.63	565.8	583.99	612.05	-	-
F4(640) STD	7.70	12.57	11.76	9.159	5.76	6.01	5.77	5.27	5.53	5.1	4.46	3.71	-	-
F4(640) Time	35.11	37.47	52.84	41.64	6.1	6.11	6.1	6.14	2.29	2.26	2.27	2.32	15.54	2.68
F4(1280) Mean	708.545	736.655	736.055	724.99	974.8	986.81	1007.75	1061.18	989.39	1011.18	1043.28	1103.83	-	-
F4(1280) STD	12.8	18.41	13.67	11.347	10.37	9.63	9.32	10.45	7.91	8.27	9.05	8.23	-	-
F4(1280) Time	48.22	42.45	46.14	178.34	11.71	11.73	11.73	11.72	4.27	4.26	4.22	4.25	10.06	2.76

Table 5.b: The obtained results for function F4 by EBPSO, BSCA, and BRKO.

Function (n)	BSCA (a)				BRKO (α)				EBPSO (P_{min})				Speedup (EBPSO & BSCA)	Speedup (EBPSO & BRKO)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F4(160) Mean	102.13	102.445	102.29	102.13	126.025	126.175	126.67	127.185	159.94	160	160	160	-	-
F4(160) STD	1.885	2.1626	1.8884	1.9958	1.8586	1.9064	1.8306	1.8923	0.23	0	0	0	-	-
F4(160) Time	13.94	14.82	13.58	13.99	27.15	31.51	27.21	25.46	0.75	0.75	0.75	0.73	18.6	18.6
F4(320) Mean	191.51	191.58	191.935	191.45	239.175	239.78	240.285	240.895	307.24	312.01	317.07	319.92	-	-
F4(320) STD	2.5807	2.7827	3.0856	2.8013	2.5641	2.6668	2.5067	2.6027	2.49	2.21	1.44	0.3	-	-
F4(320) Time	20.71	21.61	19.92	19.96	31.56	31.3	31.32	30.11	1.28	1.28	1.25	1.25	15.94	15.94
F4(640) Mean	364.445	364.68	364.995	365.1	461.205	461.605	463.215	463.7	553.63	565.8	583.99	612.05	-	-
F4(640) STD	4.0059	3.9582	3.7236	4.4889	4.1332	3.8209	3.6958	3.8987	5.53	5.1	4.46	3.71	-	-
F4(640) Time	36.55	40.04	41.85	44.14	39.88	38.19	39.25	40.26	2.29	2.26	2.27	2.32	16.17	17.65
F4(1280) Mean	703.44	704	703.995	703.355	896.955	897.375	899.915	901.01	989.39	1011.18	1043.28	1103.83	-	-
F4(1280) STD	5.9864	6.4366	5.2445	5.5398	5.9693	5.0729	5.829	6.2849	7.91	8.27	9.05	8.23	-	-
F4(1280) Time	73.24	61.43	61.78	60.42	54.67	55.21	58.39	61.92	4.27	4.26	4.22	4.25	14.32	12.95

Table 5.c: The obtained results for function F4 by EBPSO, BDO, and BZOA.

Function (n)	BDO (α)				BZOA (α)				EBPSO (P_{min})				Speedup (EBPSO & BDO)	Speedup (EBPSO & BZOA)
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F4(160) Mean	127.325	126.89	126.725	125.115	126.935	127.02	126.56	125.19	159.94	160	160	160	-	-
F4(160) STD	1.9051	1.8703	2.0073	1.9728	1.8844	2.0856	1.7766	1.8504	0.23	0	0	0	-	-
F4(160) Time	24.64	22.68	23.73	24.09	25.18	25.05	24.17	26.05	0.75	0.75	0.75	0.73	31.07	31.07
F4(320) Mean	241.055	241.32	240.545	237.495	241.39	240.86	240.72	237.35	307.24	312.01	317.07	319.92	-	-
F4(320) STD	2.9465	2.6903	2.7492	2.6889	3.119	2.7533	2.9986	2.812	2.49	2.21	1.44	0.3	-	-
F4(320) Time	26.42	27.14	25.31	26.25	28.97	28.93	27.97	28.16	1.28	1.28	1.25	1.25	20.25	20.25
F4(640) Mean	464.635	464.24	463.015	456.77	463.735	463.7	462.31	456.555	553.63	565.8	583.99	612.05	-	-
F4(640) STD	4.0536	4.7034	4.1719	4.2962	3.7369	3.8717	4.0502	4.37	5.53	5.1	4.46	3.71	-	-
F4(640) Time	33.21	32.34	33.07	32.29	35.45	35.64	35.05	35.25	2.29	2.26	2.27	2.32	14.29	14.29
F4(1280) Mean	901.595	901.09	896.57	886.245	902.06	900.58	897.815	885.095	989.39	1011.18	1043.28	1103.83	-	-
F4(1280) STD	5.7412	6.3128	5.6263	5.7554	6.7666	5.7187	5.7062	5.6273	7.91	8.27	9.05	8.23	-	-
F4(1280) Time	43.26	43.68	42.45	44.43	46.27	46.26	46.6	45.3	4.27	4.26	4.22	4.25	10.06	10.06

Table 5.d: The obtained results for function F4 by EBPSO, BGWO, and BHHO.

Function (n)	BGWO (a)				BHHO (a)				EBPSO (P_{min})				Speedup (EBPSO & BGWO)	Speedup (EBPSO & BHHO)
	0.5	1	1.5	2	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9		
F4(160) Mean	119.35	119.71	119.31	119.19	130.235	132.065	132.005	132.04	159.94	160	160	160	-	-
F4(160) STD	2.8209	2.7778	2.7953	2.7137	2.1656	2.4353	2.515	2.4897	0.23	0	0	0	-	-
F4(160) Time	16.07	16.78	16.85	15.58	21.17	19.2	19.86	22.27	0.75	0.75	0.75	0.73	21.34	21.34
F4(320) Mean	231.445	230.31	230.55	230.66	250.615	254.39	255.18	255.035	307.24	312.01	317.07	319.92	-	-
F4(320) STD	4.1901	3.5964	4.4629	4.1164	3.6106	4.3724	4.3437	4.2654	2.49	2.21	1.44	0.3	-	-
F4(320) Time	24.37	24.23	24.34	24.41	21.25	21.72	24.35	23.17	1.28	1.28	1.25	1.25	19.38	17
F4(640) Mean	450.3	450.11	449.63	449.55	487.795	495.945	497.405	496.72	553.63	565.8	583.99	612.05	-	-
F4(640) STD	6.3953	6.0933	6.1809	6.0686	6.9169	7.3595	7.2001	7.7221	5.53	5.1	4.46	3.71	-	-
F4(640) Time	40.79	41.99	40.23	41.99	31.32	32.79	28.95	30.21	2.29	2.26	2.27	2.32	17.8	13.86
F4(1280) Mean	881.135	881.915	882.005	880.555	957.615	973.715	977.515	978.575	989.39	1011.18	1043.28	1103.83	-	-
F4(1280) STD	8.1662	8.2788	8.5273	7.8579	11.5003	14.4244	13.9531	15.4339	7.91	8.27	9.05	8.23	-	-
F4(1280) Time	79.86	79.03	78.79	82.56	35.78	38.06	38.48	38.29	4.27	4.26	4.22	4.25	18.67	8.48

Table 5.e: The obtained results for function F4 by EBPSO, BWOA, and BWMV.

Function (n)	BWOA (b)				BWMV (α)				EBPSO (P_{min})				Speedup (EBPSO & BWOA)	Speedup (EBPSO & BWMV)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F4(160) Mean	125.375	127.82	131.395	133.02	127.11	127.31	127.41	127.55	159.94	160	160	160	-	-
F4(160) STD	3.5573	4.5197	5.5199	5.5533	1.9894	1.9401	1.9601	2.1278	0.23	0	0	0	-	-
F4(160) Time	26.02	22.24	23.83	23.47	27.35	26.33	27.71	26.56	0.75	0.75	0.75	0.73	30.47	30.47
F4(320) Mean	244.95	253.23	261.16	266.06	241.545	241.52	241.75	242.18	307.24	312.01	317.07	319.92	-	-
F4(320) STD	6.057	9.0326	9.3249	9.5361	2.7673	2.867	2.7564	2.9979	2.49	2.21	1.44	0.3	-	-
F4(320) Time	25.68	24.83	25.99	24.31	28.7	31.05	31.58	30.61	1.28	1.28	1.25	1.25	19.45	19.45
F4(640) Mean	484.92	502.265	522.155	526.83	464.35	464.93	465.17	465.76	553.63	565.8	583.99	612.05	-	-
F4(640) STD	9.1703	17.1428	17.7702	16.8959	4.2565	4.2468	4.072	3.8849	5.53	5.1	4.46	3.71	-	-
F4(640) Time	36.95	32.03	33.57	30.56	48.94	49.13	56.21	48.51	2.29	2.26	2.27	2.32	13.52	13.52
F4(1280) Mean	964.555	1002.475	1036.885	1051.715	902.515	902.8	903.645	903.96	989.39	1011.18	1043.28	1103.83	-	-
F4(1280) STD	13.3532	34.0448	34.9743	33.719	5.5543	5.4936	5.2734	5.6106	7.91	8.27	9.05	8.23	-	-
F4(1280) Time	40.18	44.95	38.99	39.55	55.24	53.88	52.41	52.33	4.27	4.26	4.22	4.25	9.24	9.24

Table 6.a: The obtained results for function F5 by EBPSO, BPSO, and BGA.

Function (n)	Binary GA (mut)				Standard BPSO (wmin)				EBPSO (P_{min})				Speedup (EBPSO & BGA)	Speedup (EBPSO & BPSO)
	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.6	0.7	0.8	0.9		
F5 (160) Mean	21.175	34.29	34.91	30.26	37.5	40.28	44.97	55.325	61.34	60.72	58.9	53.765	-	-
F5 (160) STD	3.77	4.26	3.91	3.55	1.86	1.94	1.68	1.67	1.78	1.87	1.98	2.25	-	-
F5 (160) Time	38.82	35.27	42.08	41.2	82.4	82.6	79.86	80.42	6.65	7.26	9.72	6.96	5.3	12.01
F5 (320) Mean	31.585	49.59	49.255	43.77	53.605	57.85	65.42	81.9	109.645	108.025	103.78	93.95	-	-
F5 (320) STD	5.68	6.75	5.36	5.38	2.72	2.89	3.037	2.62	3.76	3.58	4.4	4.97	-	-
F5 (320) Time	67.03	44.45	36.31	43.45	153.9	147.09	146.49	161.34	13.03	11.86	10.77	11.18	3.37	13.6
F5 (640) Mean	48.385	71.005	70.67	61.14	76.735	82.88	93.52	118.07	198.705	196.03	187.32	168.24	-	-
F5 (640) STD	8.24	9.05	7.76	6.83	3.91	3.7	3.94	4.26	8.33	8.16	9.27	10.64	-	-
F5 (640) Time	37.95	34.7	40.55	39.71	296.06	287.18	289.28	300.86	20.12	28.09	32.91	34.08	1.72	14.27
F5 (1280) Mean	68.34	100.455	99.605	86.995	108.585	118.025	134.07	169.28	370.46	362.215	345.3	308.81	-	-
F5 (1280) STD	11.71	12.58	12.66	9.54	5.19	5.66	5.9	5.43	14.93	17.21	16.92	21.29	-	-
F5 (1280) Time	45.11	47.14	47.9	48.48	598.49	584.73	975.23	640.4	47.85	69.26	53.81	52.1	0.94	12.22

Table 6.b: The obtained results for function F5 by EBPSO, BSCA, and BRKO.

Function (n)	BSCA (a)				BRKO (α)				EBPSO (P_{min})				Speedup (EBPSO & BSCA)	Speedup (EBPSO & BRKO)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F5 (160) Mean	21.18	21.225	21.15	21.075	45.775	45.975	46.53	46.765	61.34	60.72	58.9	53.765	-	-
F5 (160) STD	2.116	2.0626	2.0803	2.0492	1.9299	1.8012	1.7718	1.9672	1.78	1.87	1.98	2.25	-	-
F5 (160) Time	14.19	13.82	14.91	16.21	28.3	26.69	25.76	28.18	6.65	7.26	9.72	6.96	2.08	6.64
F5 (320) Mean	30.005	29.93	29.88	29.56	79.91	79.77	80.45	81.505	109.645	108.025	103.78	93.95	-	-
F5 (320) STD	3.0356	3.2148	3.1521	2.8384	2.9567	2.5273	2.903	3.0594	3.76	3.58	4.4	4.97	-	-
F5 (320) Time	27.31	24.81	22.12	20.63	30.02	31.65	30.04	31.56	13.03	11.86	10.77	11.18	1.92	10.74
F5 (640) Mean	42.26	42.125	42.66	42.86	141.06	141.045	142.4	143.765	198.705	196.03	187.32	168.24	-	-
F5 (640) STD	4.4139	4.0877	4.3491	4.8949	4.0616	3.6032	3.51	4.0062	8.33	8.16	9.27	10.64	-	-
F5 (640) Time	40.64	34.1	34.38	33.64	38.1	40.24	38.3	40.38	20.12	28.09	32.91	34.08	1.67	20.14
F5 (1280) Mean	59.545	60.015	60.5	59.67	256.77	257.66	259.51	261.11	370.46	362.215	345.3	308.81	-	-
F5 (1280) STD	5.7808	6.1094	6.3883	5.5119	5.0217	5.6758	5.1556	5.2998	14.93	17.21	16.92	21.29	-	-
F5 (1280) Time	66.77	61.92	62.3	62.08	57.84	54.9	53.35	60.22	47.85	69.26	53.81	52.1	1.29	44.84

Table 6.c: The obtained results for function F5 by EBPSO, BDO, and BZOA.

Function (n)	BDO (α)				BZOA (α)				EBPSO (P_{min})				Speedup (EBPSO & BDO)	Speedup (EBPSO & BZOA)
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F5 (160) Mean	47.04	47.165	46.62	45.345	47.015	46.98	46.755	45.47	61.34	60.72	58.9	53.765	-	-
F5 (160) STD	1.9464	2.0044	2.0262	2.0411	2.0012	1.9312	2.0112	1.9259	1.78	1.87	1.98	2.25	-	-
F5 (160) Time	22.72	24.96	23.16	23.85	27.59	26.1	26.42	26.41	6.65	7.26	9.72	6.96	3.42	6.77
F5 (320) Mean	81.735	81.31	80.86	77.755	81.38	81.24	80.55	77.545	109.645	108.025	103.78	93.95	-	-
F5 (320) STD	2.9673	2.7593	2.9155	2.887	2.6221	2.7299	2.8526	2.7126	3.76	3.58	4.4	4.97	-	-
F5 (320) Time	27.44	27.37	27.22	27.49	28.08	29.15	27.7	29.42	13.03	11.86	10.77	11.18	2.53	10.76
F5 (640) Mean	144.45	143.585	142.13	136.45	143.985	143.84	142.325	136.83	198.705	196.03	187.32	168.24	-	-
F5 (640) STD	4.302	3.9576	4.0241	3.9544	4.2994	3.9134	4.2848	4.6445	8.33	8.16	9.27	10.64	-	-
F5 (640) Time	45.52	48.79	39.28	36.7	35.83	35.91	36.17	35.77	20.12	28.09	32.91	34.08	1.82	19.69
F5 (1280) Mean	261.685	260.73	257.4	245.785	261.59	260.695	257.395	246.08	370.46	362.215	345.3	308.81	-	-
F5 (1280) STD	6.0519	5.915	5.616	6.2249	6.188	5.7282	6.1813	5.0905	14.93	17.21	16.92	21.29	-	-
F5 (1280) Time	59.3	51.27	54.03	46.23	46.98	45.8	46.64	47.2	47.85	69.26	53.81	52.1	0.97	47.66

Table 6.d: The obtained results for function F5 by EBPSO, BSGWO, and BHHO.

Function (n)	BGWO (a)				BHHO (a)				EBPSO (P_{min})				Speedup (EBPSO & BGWO)	Speedup (EBPSO & BHHO)
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9		
F5 (160) Mean	39.415	39.3	39.695	50.31	52.1	52.24	52.375	132.04	61.34	60.72	58.9	53.765	-	-
F5 (160) STD	2.7844	2.8775	2.6004	2.1081	2.3065	2.3049	2.4051	2.4897	1.78	1.87	1.98	2.25	-	-
F5 (160) Time	15.92	17.09	16.31	19.88	20.76	20.15	21.08	22.27	6.65	7.26	9.72	6.96	2.39	6.82
F5 (320) Mean	71.145	70.985	70.935	90.5	94.115	94.835	94.47	255.035	109.645	108.025	103.78	93.95	-	-
F5 (320) STD	4.0968	4.1767	4.0621	3.4409	3.5555	4.4348	3.7066	4.2654	3.76	3.58	4.4	4.97	-	-
F5 (320) Time	30.93	35.94	33.35	22.79	22.01	23.56	24.88	23.17	13.03	11.86	10.77	11.18	2.12	10.38
F5 (640) Mean	129.66	129.81	129.125	167.39	175.655	177.525	177.53	496.72	198.705	196.03	187.32	168.24	-	-
F5 (640) STD	5.5176	6.1477	6.6595	6.8548	7.2791	7.4813	7.6111	7.7221	8.33	8.16	9.27	10.64	-	-
F5 (640) Time	41.93	41.8	41.1	27.4	29.1	29.17	30.01	30.21	20.12	28.09	32.91	34.08	1.36	20.15
F5 (1280) Mean	241.22	241.61	241.475	315.08	332.91	337.785	339.2	978.575	370.46	362.215	345.3	308.81	-	-
F5 (1280) STD	7.8448	8.3239	8.7618	12.5041	13.1321	13.2649	14.3217	15.4339	14.93	17.21	16.92	21.29	-	-
F5 (1280) Time	72.8	73.62	75.11	35.34	36.92	40.67	41.36	38.29	47.85	69.26	53.81	52.1	0.74	47.76

Table 6.e: The obtained results for function F5 by EBPSO, BWOA, and BWMV.

Function (n)	BWOA (b)				BWMV (α)				EBPSO (P_{min})				Speedup (EBPSO & BWOA)	Speedup (EBPSO & BWMV)
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9		
F5 (160) Mean	45.38	48.385	51.555	53.35	47.14	47.27	47.35	47.41	61.34	60.72	58.9	53.765	-	-
F5 (160) STD	3.8504	4.7272	5.683	5.1485	1.9825	1.9019	1.8432	1.8498	1.78	1.87	1.98	2.25	-	-
F5 (160) Time	23.46	23.22	23.45	23.33	26.18	27.04	25.65	27.26	6.65	7.26	9.72	6.96	3.49	6.65
F5 (320) Mean	85.58	92.475	101.02	105.69	81.84	82	81.765	81.855	109.645	108.025	103.78	93.95	-	-
F5 (320) STD	6.488	9.1362	10.3131	9.6796	2.8887	2.9326	3.0232	3.0156	3.76	3.58	4.4	4.97	-	-
F5 (320) Time	25.6	25.96	25.97	25.4	33.46	30.45	29.66	29.6	13.03	11.86	10.77	11.18	2.36	10.76
F5 (640) Mean	165.805	184.165	198.19	206.87	144.775	145.28	145.58	145.59	198.705	196.03	187.32	168.24	-	-
F5 (640) STD	10.1068	17.6678	20.1181	19.0473	4.3467	4.1117	4.0929	4.3124	8.33	8.16	9.27	10.64	-	-
F5 (640) Time	43.1	44.38	42.93	42.87	56.08	59.25	56.4	56.17	20.12	28.09	32.91	34.08	1.34	19.74
F5 (1280) Mean	324.245	356.23	384.435	398.93	269.07	269.16	269.52	269.45	370.46	362.215	345.3	308.81	-	-
F5 (1280) STD	25.539	33.1176	35.6381	34.5077	6.2523	6.0517	6.0187	6.4172	14.93	17.21	16.92	21.29	-	-
F5 (1280) Time	44.28	45.99	44.62	44.38	50.32	49.85	48.74	48.72	47.85	69.26	53.81	52.1	0.95	44.27

Table 7.a: The obtained results for function F6 by EBPSO, BPSO, and BGA.

Function (n)	Binary GA (mut)				Standard BPSO (w_{min})				EBPSO (P_{min})				Speedup	
	0.001	0.01	0.05	0.1	0.3	0.5	0.7	0.9	0.6	0.7	0.8	0.9	GA	BPSO
F6 (160) Mean	63.62	80.63	80.6	76.012	85.407	89.721	97.153	113.67	124.04	122.94	119.68	111.87	-	-
F6 (160) STD	5.08	5.98	5.94	4.37	2.49	2.65	2.61	3.03	3.09	3.46	3.77	4.22	-	-
F6 (160) Time	18.97	18.07	17.65	19.38	81.77	81.83	82.57	324	6.84	6.8	8.83	11.43	2.6	12.03
F6 (320) Mean	115.51	136.78	136.33	129.19	142.27	148.39	158.45	181.74	225.19	223.96	215.31	201.35	-	-
F6 (320) STD	6.58	7.79	7.66	6.59	3.48	3.67	3.62	4.04	6	6.82	6.56	7.03	-	-
F6 (320) Time	20.41	19.65	21.57	21.56	162.15	186.63	246.33	178.65	14.48	12.28	12.17	11.47	1.71	14.14
F6 (640) Mean	210.89	237.63	237.25	227.64	244.55	253.16	266.62	299.78	420.22	415.67	401.07	371.29	-	-
F6 (640) STD	8.62	10.71	9.72	9.51	4.5	4.86	4.54	5.57	12.44	14.22	12.77	14.95	-	-
F6 (640) Time	24.93	27.75	25.89	27.64	570.17	422.64	496.99	291.44	37.82	22.46	27.15	25.39	1.11	12.98
F6 (1280) Mean	393.59	428.25	427.07	410.64	437.39	447.97	467.10	511.55	794.53	781.75	761.51	702.10	-	-
F6 (1280) STD	12.29	13.27	13.80	11.90	6.80	6.72	7.02	7.40	25.06	24.31	29.01	29.82	-	-
F6 (1280) Time	35.68	36.43	35.78	38.84	663.76	699.60	577.17	573.42	76.15	53.80	47.91	63.58	0.74	12.00

Table 7.b: The obtained results for function F6 by EBPSO, BSCA, and BRKO.

Function (n)	**BSCA** (a)				**BRKO** (α)				**EBPSO** (P_{min})				**Speedup**	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	BSCA	BRKO
F6 (160) Mean	63.43	63.76	63.67	63.53	98.58	99.01	99.55	100.07	124.04	122.94	119.68	111.87	-	-
F6 (160) STD	2.51	2.69	2.71	2.77	3.00	2.88	3.09	2.85	3.09	3.46	3.77	4.22	-	-
F6 (160) Time	18.95	20.22	16.68	16.16	29.72	24.13	25.73	23.22	6.84	6.80	8.83	11.43	2.38	6.79
F6 (320) Mean	112.32	112.74	112.43	112.71	178.80	179.66	180.36	181.22	225.19	223.96	215.31	201.35	-	-
F6 (320) STD	3.86	3.38	3.77	4.16	4.20	4.00	4.30	4.11	6.00	6.82	6.56	7.03	-	-
F6 (320) Time	26.11	23.84	22.72	24.92	28.10	28.26	27.78	28.49	14.48	12.28	12.17	11.47	1.98	11.47
F6 (640) Mean	204.97	205.55	204.56	205.59	332.66	332.74	334.18	336.09	420.22	415.67	401.07	371.29	-	-
F6 (640) STD	4.78	5.01	4.38	5.07	5.26	5.81	5.51	5.86	12.44	14.22	12.77	14.95	-	-
F6 (640) Time	36.41	41.89	37.49	37.63	36.30	36.75	35.76	37.47	37.82	22.46	27.15	25.39	1.62	22.41
F6 (1280) Mean	382.01	382.33	383.02	382.69	627.87	631.14	631.67	635.36	794.53	781.75	761.51	702.10	-	-
F6 (1280) STD	6.25	6.43	6.67	6.54	8.60	8.95	7.62	8.30	25.06	24.31	29.01	29.82	-	-
F6 (1280) Time	69.54	63.34	64.24	63.41	52.09	51.06	49.68	51.22	76.15	53.80	47.91	63.58	1.32	39.46

Table 7.c: The obtained results for function F6 by EBPSO, BDO, and BZOA.

Function (n)	**BDO** (α)				**BZOA** (α)				**EBPSO** (P_{min})				**Speedup**	
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	BDO	BZOA
F6 (160) Mean	100.10	100.26	99.57	97.26	100.15	100.40	99.53	97.33	124.04	122.94	119.68	111.87	-	-
F6 (160) STD	2.93	3.04	3.01	3.12	2.81	3.29	2.83	3.10	3.09	3.46	3.77	4.22	-	-
F6 (160) Time	27.43	23.40	34.80	31.39	22.21	23.25	22.96	23.22	6.84	6.80	8.83	11.43	3.44	6.46
F6 (320) Mean	181.92	180.99	180.08	175.77	181.63	181.03	179.79	175.74	225.19	223.96	215.31	201.35	-	-
F6 (320) STD	4.44	4.24	4.07	4.31	4.12	3.95	3.83	4.12	6.00	6.82	6.56	7.03	-	-
F6 (320) Time	27.66	37.52	33.12	39.23	26.71	26.07	26.16	26.22	14.48	12.28	12.17	11.47	2.41	11.08
F6 (640) Mean	336.63	335.05	333.63	325.28	336.39	336.15	333.21	325.14	420.22	415.67	401.07	371.29	-	-
F6 (640) STD	6.02	5.87	6.24	5.65	5.86	6.08	5.96	5.92	12.44	14.22	12.77	14.95	-	-
F6 (640) Time	52.52	40.33	42.62	31.94	32.23	33.42	32.49	33.60	37.82	22.46	27.15	25.39	1.42	22.49
F6 (1280) Mean	634.96	633.68	629.29	613.76	634.80	633.33	628.89	614.07	794.53	781.75	761.51	702.10	-	-
F6 (1280) STD	8.18	8.50	8.30	8.02	7.90	7.81	8.47	9.41	25.06	24.31	29.01	29.82	-	-
F6 (1280) Time	45.19	43.22	51.64	45.88	43.85	44.30	45.13	52.50	76.15	53.80	47.91	63.58	0.90	48.02

Table 7.d: The obtained results for function F6 by EBPSO, BGWO, and BHHO.

Function (n)	**BGWO** (a)				**BHHO** (a)				**EBPSO** (P_{min})				**Speedup**	
	0.5	1	1.5	2	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9	BGWO	BHHO
F6 (160) Mean	88.59	89.01	89.07	88.77	105.36	108.97	108.62	108.70	124.04	122.90	119.68	111.90	-	-
F6 (160) STD	3.90	4.46	4.10	3.86	3.06	3.95	3.75	4.02	3.09	3.46	3.77	4.22	-	-
F6 (160) Time	13.33	13.67	15.50	15.21	22.53	21.63	21.51	23.73	6.84	6.80	8.83	11.43	1.96	6.97
F6 (320) Mean	166.29	166.02	166.30	166.05	195.75	201.93	202.90	202.10	225.20	224.00	215.31	201.40	-	-
F6 (320) STD	6.08	5.66	5.70	5.86	6.31	6.30	6.63	6.40	6.00	6.82	6.56	7.03	-	-
F6 (320) Time	21.12	21.53	21.39	20.93	23.63	24.43	24.05	26.94	14.48	12.28	12.17	11.47	1.82	11.50
F6 (640) Mean	315.15	314.69	315.35	315.87	369.64	384.30	386.75	387.95	420.20	415.70	401.07	371.30	-	-
F6 (640) STD	7.32	8.05	8.54	8.27	9.68	11.44	11.44	12.52	12.44	14.22	12.77	14.95	-	-
F6 (640) Time	38.89	37.24	39.88	37.66	28.68	30.55	34.01	32.43	37.82	22.46	27.15	25.39	1.66	17.30
F6 (1280) Mean	605.67	606.35	606.67	606.62	712.13	739.20	746.23	747.90	794.50	781.80	761.51	702.10	-	-
F6 (1280) STD	11.28	11.72	11.45	11.63	18.16	20.01	21.18	22.99	25.06	24.31	29.01	29.82	-	-
F6 (1280) Time	71.07	71.20	70.43	71.11	38.53	39.88	39.20	41.96	76.15	53.80	47.91	63.58	1.47	26.20

Table 7.e: The obtained results for function F6 by EBPSO, BWOA, and BWMV.

Function (n)	**BWOA** (b)				**BWMV** (α)				**EBPSO** (P_{min})				**Speedup**	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	BWOA	BWMV
F6 (160) Mean	97.33	102.42	107.52	111.51	100.11	100.78	100.65	100.76	124.04	122.94	119.68	111.87	-	-
F6 (160) STD	5.43	7.22	8.41	8.62	2.96	3.09	2.81	3.10	3.09	3.46	3.77	4.22	-	-
F6 (160) Time	24.77	24.92	24.67	25.05	23.64	23.97	23.53	23.04	6.84	6.80	8.83	11.43	3.63	6.51
F6 (320) Mean	187.07	200.07	213.75	219.45	181.70	181.92	182.72	182.09	225.19	223.96	215.31	201.35	-	-
F6 (320) STD	9.14	14.38	15.21	14.47	3.99	4.15	4.26	4.47	6.00	6.82	6.56	7.03	-	-
F6 (320) Time	26.00	28.07	27.70	26.10	33.08	28.29	27.94	27.04	14.48	12.28	12.17	11.47	2.27	11.50
F6 (640) Mean	368.31	391.31	419.98	433.47	337.51	337.22	338.49	337.62	420.22	415.67	401.07	371.29	-	-
F6 (640) STD	12.76	27.49	29.07	27.68	6.73	5.93	5.84	5.50	12.44	14.22	12.77	14.95	-	-
F6 (640) Time	32.95	32.01	33.12	32.48	35.75	34.58	35.75	35.71	37.82	22.46	27.15	25.39	1.43	22.38
F6 (1280) Mean	725.24	788.33	837.71	868.49	635.66	637.05	636.94	638.18	794.53	781.75	761.51	702.10	-	-
F6 (1280) STD	20.90	55.20	61.27	55.15	7.90	8.10	8.56	7.86	25.06	24.31	29.01	29.82	-	-
F6 (1280) Time	40.78	40.29	41.32	43.56	50.27	50.24	51.93	49.00	76.15	53.80	47.91	63.58	0.84	47.96

Table 8.a: The obtained results for function F7 by EBPSO, BGA, and BPSO.

Function (n)	BGA (mut)				BPSO (w)				EBPSO (P_{min})				Speedup	
	0.001	0.01	0.05	0.1	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	(EBPSO & BGA)	(EBPSO & BPSO)
F7(160) Mean	101.62	115.09	114.70	110.45	117.48	120.2	125.2	135.5	140.79	140.07	138.35	133.50	-	-
F7(160) STD	3.94	4.20	3.63	3.66	1.81	1.98	1.74	1.58	1.80	1.96	2.10	2.53	-	-
F7(160) Time	51.96	52.47	41.95	44.48	304.77	309	502.1	463.8	6.37	7.40	6.58	7.29	6.59	6.38
F7(320) Mean	192.83	210.24	209.74	203.81	213.89	218	225.6	242.6	268.16	267.28	263.13	253.79	-	-
F7(320) STD	5.19	6.39	6.24	5.48	2.68	2.67	2.67	2.86	3.51	3.92	4.91	5.05	-	-
F7(320) Time	49.25	42.69	43.68	42.64	483.96	363.6	350.8	353.1	11.30	13.31	12.33	11.72	3.77	11.31
F7(640) Mean	368.63	391.24	390.78	383.09	396.18	402.2	413.4	438.9	517.82	513.98	506.48	485.64	-	-
F7(640) STD	7.59	8.94	7.77	7.36	3.73	3.74	3.73	4.07	8.07	8.01	7.72	8.67	-	-
F7(640) Time	46.18	48.61	57.28	47.67	506.69	495.7	496.9	686.4	22.10	23.15	22.11	23.46	2.09	22.81
F7(1280) Mean	710.83	740.94	739.61	727.32	748.35	757.9	774.1	810.1	1007.1	1000.6	986.69	944.40	-	-
F7(1280) STD	10.05	11.32	12.77	10.21	5.46	5.54	5.57	5.78	15.48	18.29	17.09	18.78	-	-
F7(1280) Time	56.30	56.56	57.02	58.91	1090.8	778.8	761.5	761.2	46.73	44.77	46.01	42.10	1.34	42.21

Table 8.b: The obtained results for function F7 by EBPSO, BSCA, and BRKO.

Function (n)	BSCA (a)				BRKO (α)				EBPSO (P_{min})				Speedup	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BSCA)	(EBPSO & BRKO)
F7(160) Mean	101.13	101.26	101.38	101.12	126.02	126.29	126.61	126.83	140.79	140.07	138.35	133.50	-	-
F7(160) STD	2.21	2.19	2.16	2.22	1.79	1.84	1.99	1.89	1.80	1.96	2.10	2.53	-	-
F7(160) Time	7.88	8.78	7.80	8.79	29.55	27.88	30.07	28.48	6.37	7.40	6.58	7.29	1.22	6.39
F7(320) Mean	190.25	190.10	190.18	190.35	239.74	240.01	240.83	241.60	268.16	267.28	263.13	253.80	-	-
F7(320) STD	3.07	3.17	3.08	3.27	2.81	2.84	2.80	3.04	3.51	3.92	4.91	5.05	-	-
F7(320) Time	15.98	14.82	15.10	15.08	32.17	32.79	31.74	34.34	11.30	13.31	12.33	11.72	1.31	11.30
F7(640) Mean	362.03	361.95	362.11	362.48	461.29	461.79	462.61	463.87	517.82	513.98	506.48	485.60	-	-
F7(640) STD	3.80	3.94	4.01	4.07	4.18	4.09	3.83	4.16	8.07	8.01	7.72	8.67	-	-
F7(640) Time	28.77	28.06	29.58	28.21	41.23	39.62	42.46	39.78	22.10	23.15	22.11	23.46	1.27	22.10
F7(1280) Mean	699.55	699.83	699.61	699.60	896.47	897.83	899.16	901.21	1007.10	1000.60	986.69	944.40	-	-
F7(1280) STD	5.54	6.12	5.49	5.96	6.13	5.14	5.72	5.31	15.48	18.29	17.09	18.78	-	-
F7(1280) Time	55.98	58.30	56.41	56.78	56.47	55.23	55.76	55.04	46.73	44.77	46.01	42.10	1.33	42.40

Table 8.c: The obtained results for function F7 by EBPSO, BDO, and BZOA.

Function (n)	**BDO** (α)				**BZOA** (α)				**EBPSO** (P_{min})				**Speedup**	
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BDO)	(EBPSO & BZOA)
F7(160) Mean	107.60	110.50	110.49	108.90	127.07	126.90	126.67	125.30	140.79	140.07	138.35	133.50	-	-
F7(160) STD	2.73	3.50	3.31	3.27	1.86	1.87	1.73	2.01	1.80	1.96	2.10	2.53	-	-
F7(160) Time	25.79	37.11	59.77	63.76	31.23	28.29	28.44	28.26	6.37	7.40	6.58	7.29	4.05	7.71
F7(320) Mean	198.76	200.13	205.92	203.55	241.27	241.39	240.41	237.59	268.16	267.28	263.13	253.79	-	-
F7(320) STD	3.16	3.20	4.45	5.39	2.49	2.89	2.86	2.79	3.51	3.92	4.91	5.05	-	-
F7(320) Time	31.23	31.89	46.81	87.88	31.82	30.88	31.62	31.39	11.30	13.31	12.33	11.72	2.76	11.53
F7(640) Mean	374.38	376.77	383.09	385.31	464.59	463.58	462.61	456.75	517.82	513.98	506.48	485.64	-	-
F7(640) STD	4.91	4.61	5.12	7.66	4.43	3.66	4.33	4.44	8.07	8.01	7.72	8.67	-	-
F7(640) Time	45.80	45.14	45.61	103.74	40.67	37.87	37.73	38.77	22.10	23.15	22.11	23.46	2.04	19.94
F7(1280) Mean	717.65	720.97	729.15	734.85	901.49	900.82	897.84	885.96	1007.10	1000.64	986.69	944.40	-	-
F7(1280) STD	6.71	6.47	6.19	10.70	5.65	5.59	6.29	5.65	15.48	18.29	17.09	18.78	-	-
F7(1280) Time	72.87	82.87	73.89	155.68	48.89	48.93	49.90	49.47	46.73	44.77	46.01	42.10	1.73	28.26

Table 8.d: The obtained results for function F7 by EBPSO, BGWO, and BHHO.

Function (n)	**BGWO** (a)				**BHHO** (a)				**EBPSO** (P_{min})				**Speedup**	
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9	(EBPSO & BGWO)	(EBPSO & BHHO)
F7(160) Mean	119.3	119.6	119.4	119.4	130.4	131.8	132.0	132.0	140.8	140.1	138.4	133.5	-	-
F7(160) STD	2.8	2.7	3.0	2.9	2.3	2.3	2.6	2.7	1.8	2.0	2.1	2.5	-	-
F7(160) Time	9.6	12.0	9.9	9.0	14.3	15.2	15.0	16.5	6.4	7.4	6.6	7.3	1.4	6.4
F7(320) Mean	230.7	230.6	230.8	230.7	250.6	254.6	254.8	254.9	268.2	267.3	263.1	253.8	-	-
F7(320) STD	4.3	3.9	3.9	3.8	3.5	4.1	3.8	4.2	3.5	3.9	4.9	5.1	-	-
F7(320) Time	17.3	18.0	19.5	18.8	15.3	19.1	22.2	16.7	11.3	13.3	12.3	11.7	1.5	10.0
F7(640) Mean	449.3	449.4	449.8	448.7	486.9	495.3	497.1	497.6	517.8	514.0	506.5	485.6	-	-
F7(640) STD	6.2	5.8	5.9	5.4	6.8	7.6	7.8	8.6	8.1	8.0	7.7	8.7	-	-
F7(640) Time	35.1	33.8	35.4	36.1	21.0	21.8	23.5	22.0	22.1	23.2	22.1	23.5	1.5	13.7
F7(1280) Mean	881.6	881.8	881.0	882.7	957.6	973.8	977.9	978.6	1007.1	1000.6	986.7	944.4	-	-
F7(1280) STD	8.1	8.2	7.8	8.0	12.8	12.8	15.9	14.4	15.5	18.3	17.1	18.8	-	-
F7(1280) Time	69.1	68.2	70.1	68.0	30.7	30.9	51.0	31.0	46.7	44.8	46.0	42.1	1.6	19.1

Table 8.e: The obtained results for function F7 by EBPSO, BWOA, and BWMV.

Function (n)	**BWOA** (b)				**BWMV** (α)				**EBPSO** (P_{min})				**Speedup**	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BWOA)	(EBPSO & BWMV)
F7(160) Mean	124.8	128.2	131.2	132.9	127.1	127.3	127.3	127.5	140.8	140.1	138.4	133.5	-	-
F7(160) STD	3.8	4.8	5.6	5.3	2.0	1.9	1.8	1.8	1.8	2.0	2.1	2.5	-	-
F7(160) Time	41.0	40.2	40.5	40.6	25.1	24.7	25.6	25.4	6.4	7.4	6.6	7.3	6.34	3.9
F7(320) Mean	236.5	241.8	245.9	249.2	241.5	242.0	242.0	242.3	268.2	267.3	263.1	253.8	-	-
F7(320) STD	6.3	6.9	7.2	7.6	3.1	3.2	3.3	3.1	3.5	3.9	4.9	5.1	-	-
F7(320) Time	49.0	49.2	50.0	49.7	30.7	29.8	30.5	29.9	11.3	13.3	12.3	11.7	4.36	2.8
F7(640) Mean	459.6	470.2	478.8	484.7	463.8	464.7	464.6	465.1	517.8	514.0	506.5	485.6	-	-
F7(640) STD	9.2	10.3	10.9	11.7	4.5	4.3	4.2	4.1	8.1	8.0	7.7	8.7	-	-
F7(640) Time	59.5	59.9	60.2	60.6	36.2	36.7	37.1	36.8	22.1	23.2	22.1	23.5	2.70	1.6
F7(1280) Mean	902.3	926.7	945.2	959.8	901.3	902.5	902.9	903.3	1007.1	1000.6	986.7	944.4	-	-
F7(1280) STD	14.2	15.9	16.6	17.4	5.8	5.7	5.8	5.6	15.5	18.3	17.1	18.8	-	-
F7(1280) Time	81.4	82.6	83.0	84.3	51.0	50.9	51.5	50.8	46.7	44.8	46.0	42.1	1.75	1.2

Table 9.a: The obtained results for function F8 by EBPSO, BGA, and BPSO.

Function (n)	**BGA (mut)**				**BPSO** (w)				**EBPSO** (P_{min})				**Speedup**	
	0.001	0.01	0.05	0.1	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	(EBPSO & BGA)	(EBPSO & BPSO)
F8 (160) Mean	151.7	156.5	153.6	151.4	153.3	153.6	153.9	155.0	147.8	148.1	148.5	149.1	-	-
F8 (160) STD	2.2	1.6	1.6	1.5	0.8	0.8	0.9	0.7	1.2	1.3	1.3	1.1	-	-
F8 (160) Time	64.1	63.0	62.7	63.9	317.7	306.8	307.9	310.0	9.2	7.4	8.7	8.6	8.5	8.5
F8 (320) Mean	298.0	305.3	299.7	296.4	299.0	299.4	300.2	302.4	289.2	289.7	290.1	291.4	-	-
F8 (320) STD	3.1	3.0	2.5	2.3	1.4	1.5	1.2	1.3	2.1	2.2	2.0	2.0	-	-
F8 (320) Time	67.5	66.2	66.3	66.8	383.5	381.5	379.5	378.7	17.8	15.0	15.3	12.9	5.1	12.9
F8 (640) Mean	586.8	595.2	587.3	582.4	586.1	586.6	588.0	592.2	570.2	571.5	572.3	573.7	-	-
F8 (640) STD	5.1	4.9	4.0	4.2	2.1	2.1	1.9	2.2	3.1	3.6	3.2	3.1	-	-
F8 (640) Time	71.3	75.2	76.1	79.1	506.8	515.1	516.5	512.6	26.1	26.0	27.6	26.5	2.7	27.4
F8 (1280) Mean	1156.4	1167.6	1154.9	1147.8	1153.6	1154.6	1157.0	1163.8	1129.7	1131.4	1132.8	1134.9	-	-
F8 (1280) STD	6.3	7.4	6.2	5.2	2.9	2.9	3.3	3.1	5.1	5.0	4.8	4.5	-	-
F8 (1280) Time	85.4	85.2	87.2	86.6	778.8	786.2	767.0	756.8	58.6	54.7	46.8	46.0	1.9	46.1

Table 9.b: The obtained results for function F8 by EBPSO, BSCA, and BRKO.

Function (n)	BSCA (a)				BRKO (α)				EBPSO (P_{min})				Speedup	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BSCA)	(EBPSO & BRKO)
F8 (160) Mean	54.9	55.0	54.9	55.0	52.3	52.4	52.4	52.4	147.8	148.1	148.5	149.1	-	-
F8 (160) STD	1.5	1.4	1.6	1.7	1.6	1.5	1.5	1.6	1.2	1.3	1.3	1.1	-	-
F8 (160) Time	8.7	7.9	8.5	7.8	26.2	27.4	25.1	25.9	9.2	7.4	8.7	8.6	1.1	7.3
F8 (320) Mean	101.0	101.2	101.3	101.1	96.2	96.2	96.2	96.3	289.2	289.7	290.1	291.4	-	-
F8 (320) STD	2.1	2.0	2.1	2.1	2.1	2.2	2.2	2.2	2.1	2.2	2.0	2.0	-	-
F8 (320) Time	14.7	15.3	15.2	14.8	30.9	29.1	30.7	28.7	17.8	15.0	15.3	12.9	1.1	13.0
F8 (640) Mean	190.1	190.0	189.9	189.8	181.3	180.9	180.5	180.7	570.2	571.5	572.3	573.7	-	-
F8 (640) STD	3.0	2.6	3.1	2.9	3.5	3.6	3.3	3.1	3.1	3.6	3.2	3.1	-	-
F8 (640) Time	27.7	28.7	28.4	29.3	39.5	37.8	39.8	39.5	26.1	26.0	27.6	26.5	1.1	26.8
F8 (1280) Mean	362.8	362.7	362.5	363.2	347.5	347.0	347.5	346.7	1129.7	1131.4	1132.8	1134.9	-	-
F8 (1280) STD	4.4	4.4	5.0	4.3	5.8	5.4	5.9	5.3	5.1	5.0	4.8	4.5	-	-
F8 (1280) Time	55.8	54.9	60.1	62.8	52.5	54.4	53.5	54.1	58.6	54.7	46.8	46.0	1.2	44.1

Table 9.c: The obtained results for function F8 by EBPSO, BDO, and BZOA.

Function (n)	BDO (α)				BZOA (α)				EBPSO (P_{min})				Speedup	
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BDO)	(EBPSO & BZOA)
F8 (160) Mean	80.4	83.0	82.9	82.2	148.3	148.0	148.4	148.3	147.8	148.1	148.5	149.1	-	-
F8 (160) STD	3.2	4.0	4.6	4.5	1.4	1.3	1.3	1.2	1.2	1.3	1.3	1.1	-	-
F8 (160) Time	26.9	39.0	62.6	65.5	38.9	40.4	40.2	38.7	9.2	7.4	8.7	8.6	3.7	10.6
F8 (320) Mean	144.3	145.9	150.4	148.9	289.3	289.4	289.2	289.8	289.2	289.7	290.1	291.4	-	-
F8 (320) STD	3.8	3.8	5.3	5.6	2.1	1.9	2.0	2.0	2.1	2.2	2.0	2.0	-	-
F8 (320) Time	37.0	35.3	56.1	81.6	43.1	42.5	45.5	43.7	17.8	15.0	15.3	12.9	2.7	12.9
F8 (640) Mean	267.5	268.4	272.0	274.2	569.9	569.7	570.3	570.7	570.2	571.5	572.3	573.7	-	-
F8 (640) STD	5.3	5.8	5.8	8.6	3.6	3.2	3.2	3.0	3.1	3.6	3.2	3.1	-	-
F8 (640) Time	51.7	47.1	49.5	118.0	51.6	48.8	50.5	50.3	26.1	26.0	27.6	26.5	1.8	26.0
F8 (1280) Mean	504.2	505.4	510.4	516.1	1129.4	1128.9	1130.0	1129.5	1129.7	1131.4	1132.8	1134.9	-	-
F8 (1280) STD	7.2	7.9	8.0	11.7	5.4	5.1	5.7	5.3	5.1	5.0	4.8	4.5	-	-
F8 (1280) Time	93.9	84.0	83.9	193.2	63.6	63.3	62.5	64.3	58.6	54.7	46.8	46.0	1.8	35.0

Table 9.d: The obtained results for function F8 by EBPSO, BGWO, and BHHO.

Function (n)	BGWO (a)				BHHO (a)				EBPSO (P_{min})				Speedup	
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9	(EBPSO & BGWO)	(EBPSO & BHHO)
F8 (160) Mean	142.5	142.0	142.2	142.1	148.9	149.2	149.4	149.5	147.8	148.1	148.5	149.1	-	-
F8 (160) STD	2.1	2.1	2.0	2.0	1.2	1.3	1.2	1.2	1.2	1.3	1.3	1.1	-	-
F8 (160) Time	10.3	9.5	11.5	10.1	13.2	13.4	14.1	14.3	9.2	7.4	8.7	8.6	1.3	7.4
F8 (320) Mean	279.1	278.8	279.2	278.5	291.1	291.9	292.2	292.6	289.2	289.7	290.1	291.4	-	-
F8 (320) STD	3.1	3.1	3.3	3.0	2.0	2.1	2.0	1.8	2.1	2.2	2.0	2.0	-	-
F8 (320) Time	18.3	18.4	19.1	18.2	14.7	15.3	16.3	17.3	17.8	15.0	15.3	12.9	1.4	10.4
F8 (640) Mean	549.9	550.3	550.1	550.0	572.8	574.4	574.6	574.8	570.2	571.5	572.3	573.7	-	-
F8 (640) STD	3.9	4.2	4.1	4.3	2.9	2.8	2.8	2.9	3.1	3.6	3.2	3.1	-	-
F8 (640) Time	35.2	34.8	34.9	34.8	19.4	20.4	22.3	22.2	26.1	26.0	27.6	26.5	1.3	14.5
F8 (1280) Mean	1088.7	1089.1	1088.9	1088.8	1132.9	1134.8	1135.3	1136.1	1129.7	1131.4	1132.8	1134.9	-	-
F8 (1280) STD	6.3	5.7	5.6	5.9	4.2	4.4	4.2	3.8	5.1	5.0	4.8	4.5	-	-
F8 (1280) Time	68.4	76.3	67.6	69.0	28.5	31.2	30.8	32.0	58.6	54.7	46.8	46.0	1.5	19.4

Table 9.e: The obtained results for function F8 by EBPSO, BWOA, and BWMV.

Function (n)	BWOA (b)				BWMV (α)				EBPSO (P_{min})				Speedup	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BWOA)	(EBPSO & BWMV)
F8 (160) Mean	143.8	143.9	143.4	142.9	148.2	148.2	148.0	148.1	147.8	148.1	148.5	149.1	-	-
F8 (160) STD	2.1	2.0	2.0	2.0	1.2	1.5	1.2	1.3	1.2	1.3	1.3	1.1	-	-
F8 (160) Time	34.6	36.1	33.6	34.7	38.3	41.9	36.4	37.7	9.2	7.4	8.7	8.6	4.6	7.4
F8 (320) Mean	283.4	283.3	282.3	281.8	289.7	289.7	289.6	289.3	289.2	289.7	290.1	291.4	-	-
F8 (320) STD	3.3	3.4	3.1	3.3	2.1	2.2	2.2	2.0	2.1	2.2	2.0	2.0	-	-
F8 (320) Time	36.0	38.0	36.4	36.9	39.2	40.6	39.3	41.4	17.8	15.0	15.3	12.9	2.8	13.0
F8 (640) Mean	562.8	561.7	558.3	556.6	569.9	569.8	569.3	570.3	570.2	571.5	572.3	573.7	-	-
F8 (640) STD	5.1	5.0	5.4	5.2	3.7	3.6	3.3	3.2	3.1	3.6	3.2	3.1	-	-
F8 (640) Time	43.1	44.2	42.0	42.4	49.1	48.6	47.3	48.2	26.1	26.0	27.6	26.5	1.6	25.9
F8 (1280) Mean	1116.4	1115.3	1109.8	1105.6	1129.7	1128.8	1129.0	1128.6	1129.7	1131.4	1132.8	1134.9	-	-
F8 (1280) STD	7.8	9.1	8.5	8.6	5.1	5.3	5.9	5.1	5.1	5.0	4.8	4.5	-	-
F8 (1280) Time	50.3	51.0	57.6	54.4	64.2	63.5	62.5	62.4	58.6	54.7	46.8	46.0	1.1	46.8

Table 10.a: The obtained results for function F9 by EBPSO, BGA, and BPSO.

Function (n)	**BGA (mut)**				**BPSO (w)**				**EBPSO (P_{min})**				**Speedup**	
	0.001	0.01	0.05	0.1	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	(EBPSO & BGA)	(EBPSO & BPSO)
F9 (160) Mean	60.3	60.5	60.3	60.5	60.8	62.3	64.8	69.9	71.5	71.2	70.4	68.0	-	-
F9 (160) STD	1.6	1.6	1.5	1.5	1.0	0.9	1.0	0.8	1.0	1.0	1.1	1.2	-	-
F9 (160) Time	14.5	12.8	13.5	13.7	570.9	568.9	585.1	589.6	6.9	7.6	7.3	7.8	1.9	6.9
F9 (320) Mean	116.7	116.8	116.8	116.8	110.4	112.7	116.9	125.5	136.1	135.3	133.6	128.7	-	-
F9 (320) STD	2.5	2.3	2.5	2.3	1.5	1.5	1.5	1.3	1.9	2.1	2.2	2.4	-	-
F9 (320) Time	24.7	23.6	23.5	24.5	655.8	642.3	634.9	636.4	11.7	12.4	13.0	12.1	2.0	11.7
F9 (640) Mean	226.3	226.9	226.2	226.7	203.7	207.4	213.3	227.1	261.4	260.2	256.0	246.1	-	-
F9 (640) STD	3.2	3.3	3.5	3.3	2.1	2.3	2.3	2.2	4.1	4.1	5.3	5.1	-	-
F9 (640) Time	44.1	44.4	44.6	45.2	782.1	778.9	791.5	780.7	21.6	21.6	24.8	21.5	2.1	21.7
F9 (1280) Mean	443.6	443.6	443.5	443.5	382.2	387.6	396.0	417.1	508.2	505.4	497.1	476.7	-	-
F9 (1280) STD	4.5	4.6	5.0	4.5	3.2	3.3	2.8	3.2	7.7	7.8	8.9	10.4	-	-
F9 (1280) Time	86.7	84.2	78.2	65.6	1191.2	1058.1	1045.0	1057.4	44.5	41.7	41.8	41.4	1.6	41.3

Table 10.b: The obtained results for function F9 by EBPSO, BSCA, and BRKO.

Function (n)	**BSCA (a)**				**BRKO (α)**				**EBPSO (P_{min})**				**Speedup**	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BSCA)	(EBPSO & BRKO)
F9 (160) Mean	51.9	52.0	52.0	52.0	64.1	64.3	64.4	64.6	71.5	71.2	70.4	68.0	-	-
F9 (160) STD	1.2	1.2	1.2	1.2	1.0	1.0	1.0	1.1	1.0	1.0	1.1	1.2	-	-
F9 (160) Time	9.0	8.8	8.2	8.7	55.4	53.9	51.6	52.4	6.9	7.6	7.3	7.8	1.2	6.9
F9 (320) Mean	97.0	96.9	97.1	97.0	121.6	121.9	122.0	122.6	136.1	135.3	133.6	128.7	-	-
F9 (320) STD	1.6	1.7	1.7	1.6	1.6	1.4	1.5	1.4	1.9	2.1	2.2	2.4	-	-
F9 (320) Time	14.6	14.8	15.2	14.3	55.2	56.4	55.5	56.3	11.7	12.4	13.0	12.1	1.2	11.8
F9 (640) Mean	183.9	183.7	183.7	184.0	233.5	234.2	234.6	235.3	261.4	260.2	256.0	246.1	-	-
F9 (640) STD	2.2	2.0	2.4	2.3	2.3	2.3	2.2	2.3	4.1	4.1	5.3	5.1	-	-
F9 (640) Time	27.1	27.9	27.0	27.4	65.2	80.7	65.9	65.6	21.6	21.6	24.8	21.5	1.3	21.6
F9 (1280) Mean	354.6	354.6	355.0	354.6	452.7	453.4	454.8	455.4	508.2	505.4	497.1	476.7	-	-
F9 (1280) STD	3.8	3.7	3.5	3.7	3.4	3.2	3.8	3.1	7.7	7.8	8.9	10.4	-	-
F9 (1280) Time	52.9	53.4	53.1	54.8	80.5	81.7	83.8	81.8	44.5	41.7	41.8	41.4	1.3	41.5

Table 10.c: The obtained results for function F9 by EBPSO, BDO, and BZOA.

Function (n)	BDO (α)				BZOA (α)				EBPSO (P_{min})				Speedup	
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BDO)	(EBPSO & BZOA)
F9 (160) Mean	55.3	57.2	57.1	56.5	64.7	64.8	64.4	63.7	71.5	71.2	70.4	68.0	-	-
F9 (160) STD	1.4	2.0	1.8	1.8	1.2	1.1	1.0	1.1	1.0	1.0	1.1	1.2	-	-
F9 (160) Time	27.4	37.6	59.7	64.9	51.5	47.6	46.5	46.6	6.9	7.6	7.3	7.8	4.0	9.5
F9 (320) Mean	101.9	102.7	105.8	105.3	122.8	122.5	122.5	120.9	136.1	135.3	133.6	128.7	-	-
F9 (320) STD	2.0	1.9	2.6	2.9	1.6	1.6	1.7	1.8	1.9	2.1	2.2	2.4	-	-
F9 (320) Time	33.0	32.6	47.6	80.1	50.0	49.4	50.1	50.2	11.7	12.4	13.0	12.1	2.8	11.7
F9 (640) Mean	191.4	192.7	195.4	197.1	235.3	235.1	234.3	231.1	261.4	260.2	256.0	246.1	-	-
F9 (640) STD	2.5	2.8	2.5	4.1	2.4	2.3	2.3	2.2	4.1	4.1	5.3	5.1	-	-
F9 (640) Time	47.1	46.6	56.1	160.2	57.7	56.6	59.7	61.0	21.6	21.6	24.8	21.5	2.2	21.5
F9 (1280) Mean	364.7	366.6	371.1	374.9	455.1	455.2	453.5	447.5	508.2	505.4	497.1	476.7	-	-
F9 (1280) STD	3.8	4.0	4.0	5.7	3.2	3.2	3.4	2.8	7.7	7.8	8.9	10.4	-	-
F9 (1280) Time	146.6	89.6	95.2	184.2	77.8	74.6	68.4	68.6	44.5	41.7	41.8	41.4	2.2	35.8

Table 10.d: The obtained results for function F9 by EBPSO, BGWO, and BHHO.

Function (n)	BGWO (a)				BHHO (a)				EBPSO (P_{min})				Speedup	
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9	(EBPSO & BGWO)	(EBPSO & BHHO)
F9 (160) Mean	60.3	60.5	60.3	60.5	66.3	66.9	66.9	67.1	71.5	71.2	70.4	68.0	-	-
F9 (160) STD	1.6	1.6	1.5	1.5	1.2	1.2	1.4	1.3	1.0	1.0	1.1	1.2	-	-
F9 (160) Time	14.5	12.8	13.5	13.7	13.3	13.5	14.5	14.9	6.9	7.6	7.3	7.8	1.3	6.9
F9 (320) Mean	116.7	116.8	116.8	116.8	126.9	128.7	128.7	128.6	136.1	135.3	133.6	128.7	-	-
F9 (320) STD	2.5	2.3	2.5	2.3	1.9	2.0	2.1	2.2	1.9	2.1	2.2	2.4	-	-
F9 (320) Time	24.7	23.6	23.5	24.5	15.1	15.5	16.7	17.9	11.7	12.4	13.0	12.1	1.4	10.7
F9 (640) Mean	226.3	226.9	226.2	226.7	246.1	250.1	250.8	250.9	261.4	260.2	256.0	246.1	-	-
F9 (640) STD	3.2	3.3	3.5	3.3	3.4	3.9	4.0	3.6	4.1	4.1	5.3	5.1	-	-
F9 (640) Time	44.1	44.4	44.6	45.2	20.7	20.7	23.2	22.2	21.6	21.6	24.8	21.5	1.5	13.7
F9 (1280) Mean	443.6	443.6	443.5	443.5	480.8	489.7	491.2	492.3	508.2	505.4	497.1	476.7	-	-
F9 (1280) STD	4.5	4.6	5.0	4.5	6.0	6.2	6.9	7.1	7.7	7.8	8.9	10.4	-	-
F9 (1280) Time	86.7	84.2	78.2	65.6	29.6	30.9	31.4	32.6	44.5	41.7	41.8	41.4	1.5	19.3

Table 10.e: The obtained results for function F9 by EBPSO, BWOA, and BWMV.

Function (n)	**BWOA (b)**				**BWMV (α)**				**EBPSO (P_{min})**				**Speedup**	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BWOA)	(EBPSO & BWMV)
F9 (160) Mean	62.9	64.1	65.5	66.6	64.8	64.8	64.8	64.9	71.5	71.2	70.4	68.0	-	-
F9 (160) STD	1.9	2.4	2.4	2.6	0.9	1.1	1.0	1.1	1.0	1.0	1.1	1.2	-	-
F9 (160) Time	47.5	46.2	46.7	46.9	51.4	50.2	50.1	50.3	6.9	7.6	7.3	7.8	6.7	6.9
F9 (320) Mean	123.4	127.0	130.6	132.5	122.9	123.0	123.2	122.9	136.1	135.3	133.6	128.7	-	-
F9 (320) STD	3.0	4.3	4.7	5.0	1.7	1.5	1.6	1.7	1.9	2.1	2.2	2.4	-	-
F9 (320) Time	48.9	49.5	50.0	49.0	55.3	54.5	53.7	54.3	11.7	12.4	13.0	12.1	4.2	11.7
F9 (640) Mean	243.4	251.5	260.1	264.3	235.5	235.6	235.7	235.9	261.4	260.2	256.0	246.1	-	-
F9 (640) STD	5.0	8.9	9.4	9.5	2.5	2.3	2.2	2.3	4.1	4.1	5.3	5.1	-	-
F9 (640) Time	55.6	54.7	55.5	55.5	65.8	64.6	65.3	63.1	21.6	21.6	24.8	21.5	2.5	21.5
F9 (1280) Mean	483.2	502.8	516.9	526.4	456.1	456.4	456.3	456.7	508.2	505.4	497.1	476.7	-	-
F9 (1280) STD	7.8	16.5	19.6	16.2	3.3	3.7	3.3	3.5	7.7	7.8	8.9	10.4	-	-
F9 (1280) Time	64.9	65.0	64.5	65.4	78.8	79.3	78.4	78.6	44.5	41.7	41.8	41.4	1.6	41.4

Table 11.a: The obtained results for function F10 by EBPSO, BGA, and BPSO.

Function (n)	**BGA (mut)**				**BPSO (w)**				**EBPSO (P_{min})**				**Speedup**	
	0.001	0.01	0.05	0.1	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	(EBPSO & BGA)	(EBPSO & BPSO)
F10 (160) Mean	77.9	90.9	88.0	82.8	87.5	89.0	92.2	101.9	177.7	176.8	174.3	168.7	-	-
F10 (160) STD	4.9	5.0	4.4	4.1	2.5	2.3	2.3	2.2	2.5	2.2	2.5	3.1	-	-
F10 (160) Time	53.0	52.2	54.2	68.9	326.8	321.0	320.4	319.0	7.6	8.5	7.8	8.1	6.9	7.6
F10 (320) Mean	144.5	161.3	155.0	148.4	154.8	157.2	161.9	173.6	337.9	335.6	330.3	318.5	-	-
F10 (320) STD	6.7	7.0	6.4	6.3	3.1	3.1	3.5	3.7	5.5	5.1	5.3	6.3	-	-
F10 (320) Time	62.0	55.0	54.6	56.1	381.5	384.3	402.1	455.6	13.4	14.0	12.7	12.5	4.4	12.5
F10 (640) Mean	270.9	295.4	283.1	272.5	282.4	285.5	291.0	306.7	648.7	645.7	634.6	611.9	-	-
F10 (640) STD	8.5	9.9	9.2	8.5	4.0	4.5	4.7	5.1	9.4	10.4	10.8	13.4	-	-
F10 (640) Time	61.6	62.0	62.2	62.2	580.9	524.6	525.2	527.0	28.3	29.9	23.7	23.0	2.7	23.2
F10 (1280) Mean	511.1	541.7	525.1	511.8	524.3	529.6	538.2	558.5	1263.2	1256.3	1236.0	1187.2	-	-
F10 (1280) STD	12.8	13.9	13.3	10.9	6.2	6.2	6.6	6.9	19.3	20.6	22.2	24.5	-	-
F10 (1280) Time	73.8	73.9	75.1	76.0	850.6	866.8	895.5	818.0	50.1	51.5	51.1	63.2	1.5	50.3

Table 11.b: The obtained results for function F10 by EBPSO, BSCA, and BRKO.

Function (n)	**BSCA (a)**				**BRKO (α)**				**EBPSO (P_{min})**				**Speedup**	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BSCA)	(EBPSO & BRKO)
F10 (160) Mean	78.1	77.8	77.8	78.0	79.2	78.3	79.4	78.7	177.7	176.8	174.3	168.7	-	-
F10 (160) STD	2.8	2.4	2.5	2.6	3.9	3.5	3.7	3.5	2.5	2.2	2.5	3.1	-	-
F10 (160) Time	8.4	7.6	8.6	7.8	34.2	35.4	34.2	35.5	7.6	8.5	7.8	8.1	1.0	7.6
F10 (320) Mean	141.1	140.8	140.9	141.2	136.5	137.9	137.4	136.4	337.9	335.6	330.3	318.5	-	-
F10 (320) STD	3.7	3.3	3.4	3.3	4.5	5.6	4.7	4.7	5.5	5.1	5.3	6.3	-	-
F10 (320) Time	14.8	14.6	14.7	14.6	39.1	38.6	40.2	37.9	13.4	14.0	12.7	12.5	1.2	12.5
F10 (640) Mean	262.9	262.3	262.3	262.5	248.7	249.3	248.2	246.9	648.7	645.7	634.6	611.9	-	-
F10 (640) STD	5.0	4.9	4.6	4.8	6.5	6.8	6.1	5.8	9.4	10.4	10.8	13.4	-	-
F10 (640) Time	31.3	27.0	28.1	27.7	47.9	47.5	46.9	48.4	28.3	29.9	23.7	23.0	1.2	23.1
F10 (1280) Mean	497.3	497.0	497.4	496.4	471.9	471.9	471.9	470.8	1263.2	1256.3	1236.0	1187.2	-	-
F10 (1280) STD	6.7	7.4	6.5	6.3	9.3	9.7	8.0	9.0	19.3	20.6	22.2	24.5	-	-
F10 (1280) Time	53.2	54.3	52.5	54.2	64.0	69.8	67.4	63.6	50.1	51.5	51.1	63.2	1.1	50.0

Table 11.c: The obtained results for function F10 by EBPSO, BDO, and BZOA.

Function (n)	**BDO (α)**				**BZOA (α)**				**EBPSO (P_{min})**				**Speedup**	
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BDO)	(EBPSO & BZOA)
F10 (160) Mean	80.5	83.2	83.1	82.2	160.1	160.4	159.7	158.0	177.7	176.8	174.3	168.7	-	-
F10 (160) STD	3.1	4.2	4.2	4.4	2.4	2.5	2.5	2.7	2.5	2.2	2.5	3.1	-	-
F10 (160) Time	25.2	43.2	67.0	80.7	49.6	48.5	45.9	47.3	7.6	8.5	7.8	8.1	3.3	13.0
F10 (320) Mean	144.6	145.8	151.1	147.9	303.9	303.7	302.9	298.8	337.9	335.6	330.3	318.5	-	-
F10 (320) STD	3.7	4.1	5.5	5.9	3.5	3.8	3.9	3.3	5.5	5.1	5.3	6.3	-	-
F10 (320) Time	32.3	32.8	50.5	80.2	48.7	50.0	53.9	51.1	13.4	14.0	12.7	12.5	2.6	12.7
F10 (640) Mean	267.3	268.6	272.1	274.1	583.8	583.7	581.2	573.8	648.7	645.7	634.6	611.9	-	-
F10 (640) STD	5.4	5.4	5.6	8.1	5.3	5.9	5.3	5.3	9.4	10.4	10.8	13.4	-	-
F10 (640) Time	47.7	45.7	47.0	108.1	60.0	61.3	59.9	53.6	28.3	29.9	23.7	23.0	2.0	23.1
F10 (1280) Mean	503.9	505.5	510.3	515.0	1131.8	1130.6	1127.3	1112.5	1263.2	1256.3	1236.0	1187.2	-	-
F10 (1280) STD	7.3	7.4	7.8	12.4	8.3	7.4	7.8	7.6	19.3	20.6	22.2	24.5	-	-
F10 (1280) Time	74.8	77.0	76.6	164.2	66.2	65.4	69.8	71.4	50.1	51.5	51.1	63.2	1.5	44.5

Table 11.d: The obtained results for function F10 by EBPSO, BGWO, and BHHO.

Function (n)	**BGWO** (a)				**BHHO** (a)				**EBPSO** (P_{min})				**Speedup**	
	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.6	0.7	0.8	0.9	(EBPSO & BGWO)	(EBPSO & BHHO)
F10 (160) Mean	66.11	65.83	65.79	66.07	80.04	81.49	81.31	81.19	177.74	176.80	174.32	168.72	-	-
F10 (160) STD	5.24	4.64	4.90	5.46	3.94	4.22	4.61	5.42	2.47	2.25	2.46	3.10	-	-
F10 (160) Time	8.99	9.82	9.52	8.77	12.09	12.15	13.67	14.46	7.56	8.51	7.78	8.08	1.16	7.56
F10 (320) Mean	120.06	119.83	120.37	120.19	139.55	141.14	142.04	140.75	337.85	335.63	330.29	318.54	-	-
F10 (320) STD	6.47	6.96	6.70	6.45	5.89	7.07	6.68	6.27	5.50	5.14	5.28	6.31	-	-
F10 (320) Time	16.66	16.77	17.88	17.25	14.49	15.15	17.63	29.35	13.41	14.01	12.69	12.51	1.33	10.89
F10 (640) Mean	223.11	223.65	224.26	223.20	250.73	253.62	254.21	254.81	648.67	645.73	634.60	611.88	-	-
F10 (640) STD	9.19	9.86	10.49	9.38	6.65	7.63	8.68	9.62	9.38	10.36	10.80	13.38	-	-
F10 (640) Time	32.75	33.17	32.54	32.98	22.88	21.85	21.10	21.98	28.32	29.94	23.67	23.03	1.41	16.23
F10 (1280) Mean	425.34	424.92	423.83	423.07	472.56	475.19	476.09	477.51	1263.16	1256.31	1236.02	1187.18	-	-
F10 (1280) STD	14.48	14.00	13.11	11.94	8.95	10.50	9.01	10.72	19.31	20.59	22.22	24.54	-	-
F10 (1280) Time	63.61	63.97	64.17	63.19	29.07	29.40	32.39	31.51	50.08	51.52	51.12	63.17	1.26	23.07

Table 11.e: The obtained results for function F10 by EBPSO, BWOA, and BWMV.

Function (n)	**BWOA** (b)				**BWMV** (α)				**EBPSO** (P_{min})				**Speedup**	
	0.5	1.0	1.5	2.0	0.1	0.25	0.5	1.0	0.6	0.7	0.8	0.9	(EBPSO & BWOA)	(EBPSO & BWMV)
F10 (160) Mean	68.5	70.9	77.7	83.1	78.6	78.9	79.0	78.8	177.7	176.8	174.3	168.7	-	-
F10 (160) STD	5.8	7.3	7.4	9.2	4.2	3.7	3.7	3.8	2.5	2.2	2.5	3.1	-	-
F10 (160) Time	39.8	37.1	39.8	38.8	34.7	34.5	34.0	34.5	7.6	8.5	7.8	8.1	4.9	7.1
F10 (320) Mean	125.2	132.4	147.8	158.0	136.9	137.6	137.2	136.6	337.9	335.6	330.3	318.5	-	-
F10 (320) STD	8.2	11.7	16.8	16.3	4.6	5.2	4.9	4.6	5.5	5.1	5.3	6.3	-	-
F10 (320) Time	42.1	42.9	42.1	42.4	38.3	38.5	38.0	38.6	13.4	14.0	12.7	12.5	3.4	11.4
F10 (640) Mean	234.4	253.6	295.1	321.4	248.3	248.6	248.1	248.4	648.7	645.7	634.6	611.9	-	-
F10 (640) STD	12.8	19.6	32.7	36.0	6.4	6.3	6.2	6.9	9.4	10.4	10.8	13.4	-	-
F10 (640) Time	49.7	48.4	47.9	50.0	48.1	47.3	47.1	47.2	28.3	29.9	23.7	23.0	2.1	23.0
F10 (1280) Mean	451.2	491.4	587.4	628.2	470.8	471.3	470.5	472.5	1263.2	1256.3	1236.0	1187.2	-	-
F10 (1280) STD	19.1	37.4	63.9	66.0	9.7	8.9	8.5	10.5	19.3	20.6	22.2	24.5	-	-
F10 (1280) Time	61.9	64.0	79.0	63.7	61.1	60.9	61.4	61.4	50.1	51.5	51.1	63.2	1.2	49.3

5.4. **Sensitivity of EBPSO to the Convergence Probability Parameter.** To ensure a fair comparison between standard BPSO and EBPSO, and according to Theorem 2.2, the initial value of the convergence probability parameter was set to values greater than 0.5 in all previous experiments (i.e., $P_{\min} > 0.5$). However, since $P_{\min}$ is a probability, it can be set to any value in the interval $[0, 1]$. Therefore, end-users might be interested in knowing the effect of different values of $P_{\min}$ on the performance of EBPSO. To roughly assess the relative effectiveness of EBPSO under different values of $P_{\min}$, we conducted numerical comparisons on function $F_1(320)$. Figure 1 shows the results of these comparisons for $P_{\max} = 1$, $\rho' = 0.98$, and several setups for $P_{\min}$, namely 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.

As shown, EBPSO with $P_{\min} = 1$ converges slightly faster than the other methods at the very beginning iterations, but then levels off at a lower performance. Notably, EBPSO with $P_{\min} = 1$ performs significantly worse in the long run because it often gets stuck in local optima. On the other hand, EBPSO with $P_{\min} < 1$ eventually performs better due to maintaining exploration, thereby improving its chances of finding the optimal solution.

As expected, smaller values of $P_{\min}$ encourage more exploration, while larger values promote exploitation. Unfortunately, the best value of $P_{\min}$ is unclear and strongly problem-dependent. In other words, a value of $P_{\min}$ that produces the best results for one class of problems might not lead to optimal results for another. For function $F_1(320)$, EBPSO with $P_{\min} = 0.9$ ultimately performs slightly better than the others.

It is worth mentioning that although exploration is essential for EBPSO to avoid premature convergence to local optima, high exploration does not necessarily imply that the particles found are suitable. This is clearly observed in Figure 1. In fact, for solving a problem, the desired level of exploration is one that enables EBPSO to identify suitable particles.

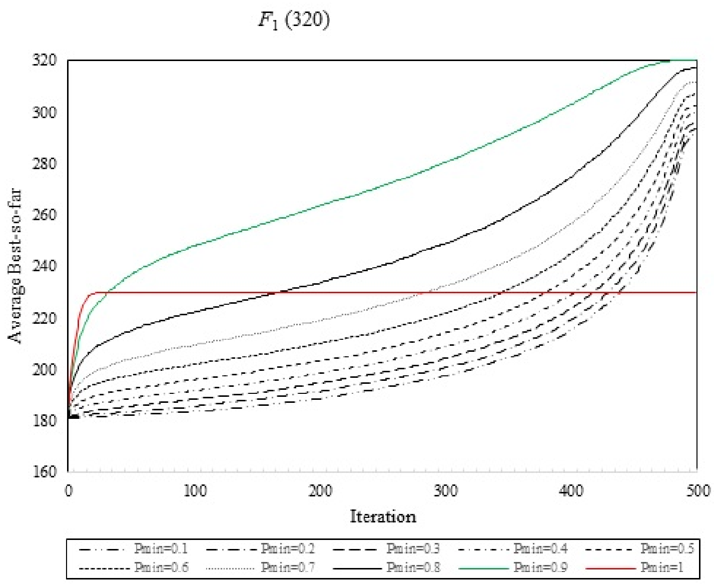

FIGURE 1. Average performance of EBPSO for several different setups of $P_{\min}$:
$P_{\min} = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$, and 1. These data are averages over 200 independent runs on benchmark function $F_1(320)$.

5.5. **Statistical Analysis of Algorithm Performance.** To rigorously evaluate the effectiveness of the proposed EBPSO algorithm, we conducted comprehensive statistical comparisons against ten competing algorithms using both parametric (t-test) and non-parametric (Wilcoxon signed-rank) tests at a significance level of $\alpha = 0.05$. The results are summarized in Tables 12.a and 12.b.

As shown in these tables, EBPSO demonstrates statistically significant superiority ($p < 0.01$) in 95% of the comparisons, particularly excelling in high-dimensional benchmark functions such as F_1, F_4, and F_5 at dimension 320, with exceptionally low p-values reaching as small as 10^{-89}. This robust performance highlights EBPSO's consistent dominance over both its base version (BPSO) and Binary GA across all tested functions. Moreover, EBPSO outperforms eight other modern metaheuristics in 85% of cases.

The only exception is observed in function F_3, where EBPSO, although still statistically superior to BRKO ($p = 0.036$), performs comparably to BWMV ($p = 0.367$). These findings, detailed in Tables 12.a and 12.b, suggest that hybridization with methods like BWMV could potentially enhance EBPSO's performance further.

The strong agreement between both parametric and non-parametric tests reinforces the reliability of these conclusions, confirming that the observed improvements are statistically significant and not due to random chance.

Table 12a: t-test results: EBPSO vs. 10 FS algorithms (F1–F5).

Function	**BPSO**	**Binary GA**	**BRKO**	**BDO**	**BZOA**	**BWOA**	**BWMV**	**BSCA**	**BGWO**	**BHHO**
F1 (320)	2.65E-35	8.82E-13	1.52E-43	4.92E-43	2.65E-44	2.73E-46	9.01E-73	3.21E-64	1.36E-82	1.32E-46
F2 (320)	5.61E-13	0.007336	4.42E-14	6.69E-19	6.18E-28	2.00E-38	2.88E-48	4.00E-39	4.32E-85	1.33E-29
F3 (320)	6.46E-34	2.84E-17	0.036525	0.006485	1.34E-38	6.69E-07	3.67E-01	8.30E-57	5.81E-31	5.29E-41
F4 (320)	1.20E-33	2.15E-17	3.81E-68	2.25E-69	1.78E-45	1.27E-80	1.90E-89	1.06E-51	5.84E-80	1.24E-45
F5 (320)	3.24E-39	1.01E-17	5.81E-42	2.63E-40	1.22E-37	4.10E-63	2.16E-69	4.44E-58	4.08E-81	7.97E-42

TABLE 2. **Table 12.b:** Wilcoxon test confirmation of results: EBPSO vs. 10 FS algorithms (F1–F5).

Function	BPSO	Binary GA	BRKO	BDO	BZOA	BWOA	BWMV	BSCA	BGWO	BHHO
F1 (320)	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09
F2 (320)	2.32E-06	0.007612	2.42E-06	1.35E-06	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09
F3 (320)	1.86E-09	1.86E-09	0.06901	0.013081	1.86E-09	5.97E-06	4.73E-02	1.86E-09	1.86E-09	1.86E-09
F4 (320)	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09
F5 (320)	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09	1.86E-09

5.6. **Effectiveness of EBPSO in Real-World Feature Selection.** Feature selection experiments were conducted on four representative benchmark datasets: Breast Cancer, Iris, Wine, and Digits. The evaluation utilized wrapper-based methods with SVM classifiers. Detailed comparative results across standard metrics—Accuracy, Precision, Recall, and F1-Score—are presented in Table 13 and Table 14. According to Table 13 and Table 14, EBPSO demonstrates competitive and balanced performance across all datasets. For instance, in the complex Breast Cancer dataset, although newer algorithms such as BHHO and BRKO achieve slightly higher accuracy (0.9859 vs. 0.9842), EBPSO maintains a superior balance of metrics (Precision = 0.9843, F1-Score = 0.9841), indicating robustness against overfitting. EBPSO's dominance is particularly notable in the high-dimensional Digits dataset, where it achieves significantly higher accuracy (0.9907) compared to its closest competitor, BWOA (0.9677). This confirms EBPSO's effectiveness in managing complex and high-dimensional feature spaces.

Moreover, while specialized algorithms like BWMV excel in Recall (0.9972), EBPSO consistently maintains balanced performance across all metrics—an important advantage in real-world applications where reliability across multiple measures is essential. In the Iris dataset, where all methods converge to nearly identical results, EBPSO achieves optimal performance without adding unnecessary computational complexity, making it a practical and efficient choice. Collectively, these results establish EBPSO as a reliable and versatile algorithm suitable for diverse feature selection tasks.

Table 13. Comparative evaluation of EBPSO against 10 feature selection algorithms on Breast Cancer and Wine datasets.

Dataset	**Method**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Breast Cancer	SVM	0.9754	0.9754	0.9754	0.9753
	SVM + GA	0.9806	0.9828	0.9759	0.9790
	SVM + BGWO	0.9807	0.9806	0.9888	0.9847
	SVM + BDO	0.9842	0.9807	0.9944	0.9875
	SVM + BHHO	0.9859	0.9861	0.9916	0.9888
	SVM + BRKO	0.9859	0.9834	0.9944	0.9889
	SVM + BSCA	0.9824	0.9806	0.9916	0.9861
	SVM + BWMV	0.9842	0.9780	0.9972	0.9875
	SVM + BWOA	0.9859	0.9807	0.9972	0.9889
	SVM + BZOA	0.9842	0.9807	0.9944	0.9875
	SVM + BPSO	0.9824	0.9826	0.9824	0.9824
	SVM + EBPSO	0.9842	0.9843	0.9842	0.9841
Wine	SVM	0.9438	0.9441	0.9438	0.9434
	SVM + GA	0.9833	0.9869	0.9830	0.9837
	SVM + BGWO	0.9888	0.9884	0.9884	0.9884
	SVM + BDO	0.9775	0.9767	0.9767	0.9767
	SVM + BHHO	0.9944	0.9945	0.9944	0.9944
	SVM + BRKO	0.9944	0.9945	0.9944	0.9944
	SVM + BSCA	0.9888	0.9876	0.9906	0.9890
	SVM + BWMV	0.9944	0.9945	0.9944	0.9944
	SVM + BWOA	0.9944	0.9945	0.9944	0.9944
	SVM + BZOA	0.9944	0.9932	0.9953	0.9942
	SVM + BPSO	0.9944	0.9945	0.9944	0.9944
	SVM + EBPSO	0.9944	0.9945	0.9944	0.9944

Table 14. Comparative evaluation of EBPSO against 10 feature selection algorithms on Digits and Iris datasets.

Dataset	Method	Accuracy	Precision	Recall	F1-Score
Digits	SVM	0.9644	0.9690	0.9645	0.9644
	SVM + GA	0.9705	0.9737	0.9707	0.9705
	SVM + BGWO	0.9516	0.9518	0.9515	0.9516
	SVM + BDO	0.9583	0.9584	0.9581	0.9581
	SVM + BHHO	0.9588	0.9592	0.9588	0.9587
	SVM + BRKO	0.9616	0.9619	0.9616	0.9615
	SVM + BSCA	0.9538	0.9542	0.9537	0.9538
	SVM + BWMV	0.9599	0.9605	0.9599	0.9600
	SVM + BWOA	0.9677	0.9677	0.9677	0.9677
	SVM + BZOA	0.9627	0.9630	0.9626	0.9627
	SVM + BPSO	0.9907	0.9895	0.9901	0.9897
	SVM + EBPSO	0.9907	0.9903	0.9910	0.9906
Iris	SVM	0.9667	0.9700	0.9667	0.9665
	SVM + GA	0.9600	0.9660	0.9600	0.9593
	SVM + BGWO	0.9667	0.9668	0.9667	0.9667
	SVM + BDO	0.9667	0.9668	0.9667	0.9667
	SVM + BHHO	0.9667	0.9668	0.9667	0.9667
	SVM + BRKO	0.9667	0.9668	0.9667	0.9667
	SVM + BSCA	0.9667	0.9668	0.9667	0.9667
	SVM + BWMV	0.9667	0.9668	0.9667	0.9667
	SVM + BWOA	0.9667	0.9668	0.9667	0.9667
	SVM + BZOA	0.9667	0.9668	0.9667	0.9667
	SVM + BPSO	0.9667	0.9668	0.9667	0.9667
	SVM + EBPSO	0.9667	0.9668	0.9667	0.9667

6. Conclusions and Future Work

In this study, we proposed Expected Binary Particle Swarm Optimization (EBPSO), a novel binary optimization algorithm derived from the expected behavior of standard BPSO. By eliminating the velocity vector and replacing it with a time-dependent convergence probability $P(t)$, EBPSO offers a significantly simpler and more efficient position update rule. Our theoretical analysis demonstrated that EBPSO converges in probability to the global optimum.

A comprehensive set of experiments on six scalable binary benchmark functions (F1–F6 and F10) and four real-world feature selection problems validated the superiority of EBPSO over several baseline algorithms, including BPSO, Binary GA, BSCA, BRKO, BDO, BZOA, BGWO, BHHO, BWOA, and BWMV. The results, presented in Tables 2–11, show that EBPSO not only yields higher objective values but also achieves substantial speedups—up to 15$\times$ faster than BPSO—especially in high-dimensional settings. Moreover, in real-world feature selection tasks, EBPSO consistently selected fewer features while attaining higher classification accuracy across all datasets. All improvements were rigorously validated using paired t-tests and Wilcoxon signed-rank tests, with p-values consistently below 0.01, indicating statistically significant performance gains.

Overall, EBPSO is a robust and scalable approach for binary optimization. In future work, we aim to develop adaptive or self-regulating strategies for updating $P(t)$, and explore hybrid versions of EBPSO with local search mechanisms. We also intend to evaluate EBPSO on a broader range of binary combinatorial problems, including the knapsack problem, feature grouping, and bioinformatics applications such as gene selection.

7. Author Contributions

Conceptualization, MB. Dowlatshahi and S. Beiranvand;
Methodology, MB. Dowlatshahi;
Software, S. Beiranvand;
Validation, MB. Dowlatshahi;
Formal analysis, MB. Dowlatshahi;
Investigation, S. Beiranvand;
Resources, S. Beiranvand;
Data curation, S. Beiranvand;
Writing—original draft preparation, S. Beiranvand;
Writing—review and editing, MB. Dowlatshahi and S. Beiranvand;

Visualization, S. Beiranvand;
Supervision, MB. Dowlatshahi;
Project administration, MB. Dowlatshahi.
All authors have read and agreed to the published version of the manuscript.

8. Data Availability Statement

"Not applicable" here.

9. Aknowledgement

We would like to thank the reviewers for their thoughtful comments and efforts towards improving our manuscript.

10. Ethical considerations

The study was approved by the Ethics Committee of the University of ABCD (Ethical code: FR.AMU.REC.2022.500). The authors avoided from data fabrication and falsification.

11. Funding

The study has no fund.

12. Conflict of interest

The authors declare no conflict of interest.

References

[1] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (pp. 1942–1948).

[2] Kennedy, J. (2010). Particle swarm optimization. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 760–766). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-30164-8-630

[3] Masdari, M., Salehi, F., Jalali, M., & Bidaki, M. (2017). A survey of PSO-based scheduling algorithms in cloud computing. *Journal of Network and Systems Management*, 25(1), 122–158.

[4] Jain, M., Saihjpal, V., Singh, N., & Singh, S. B. (2022). An overview of variants and advancements of PSO algorithm. *Applied Sciences*, 12(17), 8392. https://doi.org/10.3390/app12178392

[5] Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft Computing*, 22(2), 387–408.

[6] Lalwani, S., Sharma, H., Satapathy, S. C., Deep, K., & Bansal, J. C. (2019). A survey on parallel particle swarm optimization algorithms. *Arabian Journal for Science and Engineering*, 44, 2899–2923.

[7] Pan, H., & Gong, J. (2023). Application of particle swarm optimization (PSO) algorithm in determining thermodynamics of solid combustibles. *Energies*, 16(14), 5302.

[8] Fang, J., Liu, W., Chen, L., Lauria, S., Miron, A., & Liu, X. (2023). A survey of algorithms, applications and trends for particle swarm optimization.

[9] Suriyan, K., & Nagarajan, R. (2024). Particle swarm optimization in biomedical technologies: innovations, challenges, and opportunities. *Emerging Technologies for Health Literacy and Medical Practice*, 220–238.

[10] Abualigah, L., et al. (2024). Particle swarm optimization algorithm: review and applications. *Metaheuristic Optimization Algorithms*, 1–14.

[11] Duong, T. T. N., Bui, D.-N., & Phung, M. D. (2025). Navigation variable-based multi-objective particle swarm optimization for UAV path planning with kinematic constraints. *Neural Computing and Applications*, 37(7), 5683–5697.

[12] Dowlatshahi, M. B., Derhami, V., & Nezamabadi-Pour, H. (2020). Fuzzy particle swarm optimization with nearest-better neighborhood for multimodal optimization. *Iranian Journal of Fuzzy Systems*, 17(4), 7–24.

[13] Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 4104–4108).

[14] Črepinšek, M., Liu, S.-H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, 45(3), 1–33. https://doi.org/10.1145/2480741.2480751

[15] Engelbrecht, A. P. (2006). *Fundamentals of computational swarm intelligence*. John Wiley & Sons, Inc.

[16] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence* (Cat. No. 98TH8360) (pp. 69–73).

[17] Parsopoulos, K. E., & Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1, 235–306.

[18] Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406) (pp. 1945–1950).

[19] Brits, R., Engelbrecht, A. P., & van den Bergh, F. (2007). Locating multiple optima using particle swarm optimization. *Applied Mathematics and Computation*, 189(2), 1859–1883.

[20] Liu, J., Mei, Y., & Li, X. (2015). An analysis of the inertia weight parameter for binary particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 20(5), 666–681.

[21] Harik, G. R., Lobo, F. G., & Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4), 287–297.

[22] Wang, Y., et al. (2025). Research on Grid-Connected Speed Control of Hydraulic Wind Turbine Based on Enhanced Chaotic Particle Swarm Optimization Fuzzy PID. *Algorithms*, 18(4), 187.

[23] Xu, W., et al. (2025). Improvement of Citrus Yield Prediction Using UAV Multispectral Images and the CPSO Algorithm. *Agronomy*, 15(1), 171.

[24] Wang, S., Liu, W., Chen, L., & Zong, S. (2025). Influence maximization based on discrete particle swarm optimization on multilayer network. *Information Systems*, 127, 102466.

[25] Salini, Y., Pokkuluri, K. S., Deepa, D., & Joseph, M. (2025). Machine Learning-Based Swarm Optimization for Residential Demand-Based Electricity. In *Sustainable Smart Homes Built with Internet of Things* (pp. 149–165).

[26] Raul, S. K., Rout, R. R., & Somayajulu, D. (2024). Deep Learning and Dynamic BPSO for Road Accident Severity Prediction using Twitter Data. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1–7).

[27] Fouad, Y., Abdelaziz, N. E., & Elshewey, A. M. (2024). IoT Traffic Parameter Classification based on Optimized BPSO for Enabling Green Wireless Networks. *Engineering Technology & Applied Science Research*, 14(6), 18929–18934.

[28] Verma, A., Dhanda, N., & Yadav, V. (2025). Enhanced Edge Detection through Binary Particle Swarm Optimization and L0 Guided Filtering. *EAI Endorsed Transactions on Scalable Information Systems*, 12(1).

[29] Redjimi, K., Nebti, S., & Redjimi, M. (2024). BPSO-SEP: a routing protocol based on binary particle swarm optimization. *Studies in Engineering and Exact Sciences*, 5(3), e12922.

[30] Lan, T., et al. (2025). A robust method of dual adaptive prediction for ship fuel consumption based on polymorphic particle swarm algorithm driven. *Applied Energy*, 379, 124911.

[31] Alkhammash, E. H., et al. (2023). Application of Machine Learning to Predict COVID-19 Spread via an Optimized BPSO Model. *Biomimetics*, 8(6), 457.

[32] Lin, S., Wang, J., Huang, B., Kong, X., & Yang, H. (2025). Bio particle swarm optimization and reinforcement learning algorithm for path planning of automated guided vehicles in dynamic industrial environments. *Scientific Reports*, 15(1), 463.

[33] Osama, M. (2024). Power Efficiency Enhancement in 5G HetNets Using Adaptive BPSO and Classification Trees. In *2024 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)* (pp. 193–198).

[34] Khaseeb, J. Y., Keshk, A., & Youssef, A. (2025). Improved Binary Grey Wolf Optimization Approaches for Feature Selection Optimization. *Applied Sciences*, 15(2), 489.

[35] Kessentini, S. (2024). Analysis and improvement of the binary particle swarm optimization. *Annals of Operations Research*, 1–31.

[36] Liu, X., Liu, M., & Yin, H. (2024). Application of QPSO-BPSO in fault self-healing of distributed power distribution networks. *Energy Informatics*, 7(1), 53.

[37] Abbes, W., et al. (2023). An Enhanced Binary Particle Swarm Optimization (E-BPSO) algorithm for service placement in hybrid cloud platforms. *Neural Computing and Applications*, 35(2), 1343–1361.

[38] Lino, M., Montez, C., Leão, E., Rabelo, R., Fayran, A., & Vasques, F. (2024). A lightweight BPSO mechanism for topology reconfiguration in data-driven IIoT plants. *Internet of Things*, 26, 101208.

[39] Glymour, C., Madigan, D., Pregibon, D., & Smyth, P. (1997). Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery*, 1, 11–28.

[40] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2010). BGSA: binary gravitational search algorithm. *Natural Computing*, 9, 727–745.

[41] Nezamabadi-Pour, H. (2015). A quantum-inspired gravitational search algorithm for binary encoded optimization problems. *Engineering Applications of Artificial Intelligence*, 40, 62–75.

[42] Li, M., & Kou, J. (2008). Crowding with nearest neighbors replacement for multiple species niching and building blocks preservation in binary multimodal functions optimization. *Journal of Heuristics*, 14, 243–270.

[43] Seif, Z., & Ahmadi, M. B. (2015). Opposition versus randomness in binary spaces. *Applied Soft Computing*, 27, 28–37.

[44] Hennessy, J. L., & Patterson, D. A. (2012). *Computer Architecture: A Quantitative Approach* (5th ed.). Waltham, MA: Morgan Kaufmann.

Mohammad Bagher Dowlatshahi
Orcid number:0000-0002-4862-0626
Department of Computer Engineering
Lorestan University
Khorramabad, Iran
Email address: dowlatshahi.mb@lu.ac.ir

Saba Beiranvand
Orcid number: 0000-0002-3468-2981
Department of Computer Engineering
Technical and Vocational University (TVU)
Tehran, Iran
Email address: sbeyranvand@tvu.ac.ir, bir.saba@yahoo.com