Performance Investigation on Binary Particle Swarm Optimization for Global Optimization

Ying Loong Lee[1(✉)], Ayman Abd El-Saleh[1], Jonathan Loo[2], and MingFei Siyau[3]

[1] Faculty of Engineering, Multimedia University, Jalan Multimedia
63100, Cyberjaya, Selangor, Malaysia
lee.ying.loong12@student.mmu.edu.my, ayman.elsaleh@mmu.edu.my
[2] School of Science and Technology, Middlesex University, London NW44BT, UK
j.loo@mdx.ac.uk
[3] BiMEC Research Group, School of Engineering,
London South Bank University, London SE10AA, UK
siyaum@lsbu.ac.uk

Abstract. Binary particle swarm optimization (BinPSO) is introduced as a population-based random search algorithm for discrete binary optimization problems. A number of BinPSO variants have been introduced in the literature and showed performance improvements over the original BinPSO algorithm. However, no detailed performance comparison between these BinPSO variants has been found in the current literature. In this paper, a more thorough performance comparison study on the BinPSO variants in terms of convergence speed, solution quality and performance stability is presented. The BinPSO variants are further compared with a newly adopted cooperative BinPSO variant. The performance evaluation is conducted using De Jong's test functions, several complex multimodal functions, and a real-world engineering problem, namely optimization of the detection performance of cooperative spectrum sensing in cognitive radio networks. Results show that most of the BinPSO variants exhibit excellent performance on solving De Jong's test functions while the cooperative BinPSO variant performs better on the complex multimodal problems and the real-world engineering problem. Overall, the cooperative BinPSO variant shows the most promising performance, especially in terms of solution quality and performance stability.

Keywords: Particle swarm optimization · Binary particle swarm optimization · Convergence · Stability · De Jong's test functions · Multimodal functions

1 Introduction

Particle swarm optimization (PSO) is a population-based random search algorithm inspired from the social behavior of bird flocking [8]. Compared to other evolutionary algorithms, the implementation of PSO algorithms is relatively simple as it requires fewer parameters to adjust. In the past decade, PSO algorithms have been successfully applied to solve many real-world problems, including those requiring autonomous real-time adaptation [5, 13, 16, 18, 21].

The original PSO algorithm can only be applied to solve optimization problems that are in the continuous-valued (floating-point) domain. In order to solve discrete binary optimization problems, a binary particle swarm optimization (BinPSO) algorithm is proposed in [9]. The BinPSO algorithm retains the same advantages of the original PSO algorithm proposed in [8] and it allows for solving optimization problems of discrete binary nature. The BinPSO algorithm has been modified and implemented to solve several real-world problems [1, 14, 15, 17, 20].

In the literature, various studies have been carried out to continue improving the performance of BinPSO algorithms for global optimization. These studies have led to several new variants such as Novel BinPSO (NBinPSO) [11], Essential BinPSO (EPSO) [2] and Modified BinPSO (MBinPSO) [12]. Most of these studies employ the BinPSO algorithm proposed in [9] as the only benchmark. Comparison between the recent BinPSO variants in various performance aspects is not provided in these studies, hence leaving the progress of the BinPSO development unclear. As such, a performance comparison study on BinPSO would be useful to the readers as it would give insights about the recent development of BinPSO in various aspects.

The objective of this paper is to investigate the performance of various BinPSO variants proposed in the literature in terms of solution quality, convergence speed and performance stability. These BinPSO variants are further compared with a BinPSO algorithm enhanced with the cooperative approach in [19]. This is to investigate the potential of cooperative approaches to improve the BinPSO performance, which may spark researchers' interest to further explore into these approaches. The performance comparison is conducted using De Jong's test functions [3], which consist of several unimodal and simple multimodal optimization problems, and a number of complex multimodal functions, which contain large numbers of suboptimal solutions. In addition, a real-world application problem is used to evaluate the BinPSO variants.

The rest of the paper is organized as follows: Section 2 introduces PSO and reviews several recent BinPSO variants. In Section 3, performance evaluation of the recent BinPSO variants presented and discussed. Finally, Section 4 provides a number of concluding remarks about the performance evaluation and highlights potential future work.

2 Particle Swarm Optimization

2.1 Continuous PSO

In PSO, a group of particles travels through the search space of a given optimization problem to look for the optimal solution. This group of particles is referred to as a swarm. The position of each particle represents a candidate solution for the optimization problem. The basic working principle of the PSO algorithm is to allow all particles travelling in the search space in such a way that the movement of these particles is dependent on their personal and past experiences. In the PSO algorithm, the velocity and position update equations of each particle are given as [8]:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_1 \left(y_{ij}(t) - x_{ij}(t)\right) + c_2 r_2 \left(\hat{y}_j(t) - x_{ij}(t)\right) \tag{1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2}$$

where $x_{ij}(t)$ and $v_{ij}(t)$ are the position and velocity of the i-th particle in the j-th dimension at the t-th iteration respectively, c_1 and c_2 are acceleration constants, r_1 and r_2 are uniformly distributed random values ranging in [0, 1], $y_{ij}(t)$ is the personal best (pbest) position, which is the best position found by the i-th particle in the j-th dimension at the t-th iteration while $\hat{y}_j(t)$ is the global best (gbest) position, which is the best position found by the entire swarm in the j-th dimension at the t-th iteration.

Fig. 1 shows the PSO algorithm for solving a maximization problem, where P denotes the swarm, N_s is the swarm size, n is the number of dimensions (i.e., number of decision variables), $f(.)$ is the objective function of the maximization problem, $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$, $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \ldots, y_{in}]$ and $\hat{\boldsymbol{y}}_i = [\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{in}]$ [19]. Each particle is first initialized at a random position in the search space of the problem. In each iteration, the velocity of each particle is updated using Eq. (1) followed by the position update using Eq. (2). After the updates, the new position of each particle will be evaluated using the objective function. If the solution found by a particle in an iteration provides a better objective function value than the previous iteration, this solution will be updated as the pbest position. Similarly, if the pbest position of a particle at current iteration is better than the gbest position, the gbest position will be replaced by the pbest position. This process is repeated until a certain stopping criterion is met.

```
Create and initialize an $n$-dimensional cPSO: $P$
repeat:
    for each particle $i \in [1..N_s]$:
        if $f(P.\mathbf{x}_i) > f(P.\mathbf{y}_i)$
            then $P.\mathbf{y}_i = P.\mathbf{x}_i$
        if $f(P.\mathbf{y}_i) > f(P.\hat{\mathbf{y}}_i)$
            then $P.\hat{\mathbf{y}}_i = P.\mathbf{y}_i$
        Perform PSO updates on $P$ using Eqs. (1) and (2)
        endfor
until the stopping criterion is met
```

Fig. 1. Pseudocode of the PSO algorithm [19]

The topology of particles, i.e., their neighborhood structure can be modified to improve the performance. Various topologies have been proposed such as the global best (Gbest) model and the local best (Lbest) model [10]. In the Gbest model, all particles are neighbors to each other whereas in the Lbest model, each particle has two neighbors only. For further reading of topologies, readers may refer to [10].

2.2 Binary PSO

The first BinPSO algorithm proposed in [9] has a similar pseudocode compared to the original continuous-valued PSO algorithm except that the position update equation is replaced by:

$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } r_3 < \text{sigmoid}(v_{ij}(t)) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where r_3 is a uniformly distributed random value ranging in [0, 1] and the sigmoid function of $v_{ij}(t)$ is defined as [9]:

$$\text{sigmoid}(v_{ij}(t)) = \frac{1}{1+\exp(-v_{ij}(t))} \tag{4}$$

In the BinPSO algorithm, $x_{ij}(t)$, $y_{ij}(t)$ and $\hat{y}_j(t)$ take only the binary values; that is zero or one, while the $v_{ij}(t)$ remains continuous-valued. Unlike the PSO algorithm in [8], $v_{ij}(t)$ in the BinPSO algorithm represents the probability of $x_{ij}(t)$ approaching the value of one. Moreover, $v_{ij}(t)$ is limited by a maximum velocity, $V_{\max}$. The practical value of $V_{\max}$ was suggested to be ±4 [7]. It is noteworthy that the sigmoid function in Eq. (4) is used to normalize the $v_{ij}(t)$ so that its value is restricted within the range [0, 1].

In [11], the behavior of the parameters of the BinPSO algorithm is said to be deviated from that of the original PSO algorithm [8], leading to difficulties in choosing the appropriate parameter settings. To address these difficulties, the study in [11] proposed the NBinPSO algorithm in which a set of rules are introduced to update the position and velocity of each particle such that they imitate more closely to those of the original PSO algorithm. In addition, an inertia weight w is added to the particles' velocity to preserve the direction each of them has previously travelled.

The study in [2] exploits another approach for the same issues pointed in [11] by reformulating the particles' velocity and position update equations into probabilistic ones, deriving a new BinPSO variant known as Essential BinPSO (EPSO). Moreover, adopting the concept of ant colony optimization, a queen informant particle is added to further improve the search capability of the EPSO algorithm. As the queen informant particle will leave some pheromones in its previously travelled path in each iteration, this enhances the solution exploration and exploitation at the beginning and the end of the EPSO algorithm operation, respectively. This new EPSO variant is known as EPSOq.

In [12], the MBinPSO algorithm is proposed based on the genotype-phenotype concept where the velocity and the position of the particle are replaced by the so-called genotype and phenotype particle. The genotype particle is updated using Eqs. (1) and (2) with their respective $x_{ij}(t)$ being replaced by the phenotype and genotype particles. The phenotype particle is updated using Eqs. (3) and (4) with $x_{ij}(t)$ being replaced by the genotype particle. This modification allows the new position of each particle to take into account its previous position in the velocity and position update, which is not the case for the original BinPSO algorithm. Moreover, a bit change mutation feature is incorporated into the algorithm to mutate the genotype particle for better solution exploration.

In [19], a cooperative approach is proposed to enhance the ability of the PSO algorithm in solving high-dimensional optimization problems. This cooperative approach can be applied to any PSO algorithm, in fact, not limited to BinPSO. The idea of the cooperative approach is to decompose the candidate solution vector, which consists of all decision variables, into a number of component vectors, denoted by K. Each component vector is optimized by a separate swarm. In order to evaluate a component vector from a particular swarm, a *context vector* function is defined to compose this component vector with other component vectors from other swarms and form a global solution vector. The context vector function, $\mathbf{b}(k, \mathbf{z})$ is defined as follows [19]:

$$\mathbf{b}(k,\mathbf{z}) \equiv \left(P_1.\hat{\mathbf{y}}, ..., P_{k-1}.\hat{\mathbf{y}}, \mathbf{z}, P_{k+1}.\hat{\mathbf{y}}, ..., P_K.\hat{\mathbf{y}}\right) \tag{5}$$

where $P_k.\hat{\mathbf{y}}$ denotes the best component vector found by the k-th swarm where k = 1, 2, …, K, and $\mathbf{z}$ denotes the component vector of the k-th swarm, which is to be evaluated. Using this context vector function, a candidate solution vector is reformed which can then be evaluated by the objective function. By applying the cooperative approach to the BinPSO algorithm, the pseudocode of the cooperative BinPSO (CBinPSO) algorithm can be presented in Fig. 2 where n_b is the number of binary-valued decision variables. It is noteworthy that $K = K_1 + K_2$ swarms are created with each of the first K_1 swarms optimizing $\lceil n_b/K \rceil$ decision variables and each of the next K_2 swarms optimizing $\lfloor n_b/K \rfloor$ decision variables. K_1 and K_2 are separately calculated because n_b may not always be evenly divided by K. Then, each swarm runs the BinPSO algorithm independently and evaluates its solution vectors using Eq. (5).

```
define
$\mathbf{b}(k, \mathbf{z}) \equiv ( P_1.\hat{\mathbf{y}}, ..., P_{k-1}.\hat{\mathbf{y}}, \mathbf{z}, P_{k+1}.\hat{\mathbf{y}}, ..., P_K.\hat{\mathbf{y}})$
$K_1 = n_b \bmod K$
$K_2 = K - (n_b \bmod K)$
Initialize $K_1$ $\lceil n_b/K \rceil$-dimensional PSOs: $P_k$, $k \in [1..K_1]$
Initialize $K_2$ $\lfloor n_b/K \rfloor$-dimensional PSOs: $P_k$, $k \in [(K_1+1)..K]$
repeat:
  for each swarm $k \in [1..K]$:
    for each particle $i \in [1..N_s]$:
      if $f(\mathbf{b}(k, P_k.\mathbf{x}_i)) > f(\mathbf{b}(k, P_k.\mathbf{y}_i))$
        then $P_k.\mathbf{y}_i = P_k.\mathbf{x}_i$
      if $f(\mathbf{b}(k, P_k.\mathbf{y}_i)) > f(\mathbf{b}(k, P_k.\hat{\mathbf{y}}_i))$
        then $P_k.\hat{\mathbf{y}}_i = P_k.\mathbf{y}_i$
    Perform BinPSO updates on $P_k$ using Eqs. (1), (3) and (4)
   endfor
endfor
until the stopping condition is met
```

Fig. 2. Pseudocode of the CBinPSO algorithm

3 Performance Evaluation

The performance of various BinPSO variants is investigated in terms of solution quality, performance stability and convergence speed. The solution quality achieved by the BinPSO variants implies how well their ability in finding optimal solutions. The performance stability indicates the ability of the BinPSO variants in maintaining consistent performance in such a way that the solutions found in each run do not differ significantly compared to those found in other runs. The convergence speed shows how fast the BinPSO variants can converge to a solution. Intuitively, an effective and efficient optimization algorithm would demonstrate high solution quality, high performance stability and fast convergence speed. Using these performance aspects, the following BinPSO variants have been evaluated and compared: BinPSO [9], BinPSO [11], EPSO [2], EPSOq [2], MBinPSO [12] and CBinPSO [19].

Additionally, two different topologies, i.e., Gbest and Lbest models are applied to the BinPSO, EPSO, EPSOq and Modified BinPSO algorithms.

3.1 Benchmark Functions

In the performance evaluation, De Jong's test functions [3] and three complex multi-modal functions [12] are employed as the benchmark problems. These functions are given as follows:

$$f_1(\boldsymbol{x}) = \sum_{j=1}^{n} x_j^2 \tag{6}$$

$$f_2(\boldsymbol{x}) = \sum_{j=1}^{n-1} \left(100(x_j^2 - x_{j+1})^2 - (1 - x_j)^2\right) \tag{7}$$

$$f_3(\boldsymbol{x}) = 6n + \sum_{j=1}^{n} \lfloor x_j \rfloor \tag{8}$$

$$f_4(\boldsymbol{x}) = \sum_{j=1}^{n} j x_j^4 + U(0,1) \tag{9}$$

$$\frac{1}{f_5(\boldsymbol{x})} = \frac{1}{500} + \sum_{m=1}^{25} \frac{1}{m + \sum_{j=1}^{2} \left(x_j - a_{jm}\right)^6} \tag{10}$$

$$f_6(\boldsymbol{x}) = 10n + \sum_{j=1}^{n} \left(x_j^2 - 10\sin\left(2\pi x_j\right)\right) \tag{11}$$

$$f_7(\boldsymbol{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{j=1}^{n} x_j^2}\right) - \exp\left(\frac{1}{n}\sum_{j=1}^{n} \cos(2\pi x_j)\right) + 20 + e \tag{12}$$

$$f_8(\boldsymbol{x}) = \frac{1}{4000}\sum_{j=1}^{n} x_j^2 - \prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) + 1 \tag{13}$$

where $a_{jm} = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots \end{pmatrix}$ and n is the number of dimensions. Also, it is noteworthy that f_6 - f_8 are minimization problems. To show evaluation consistency with De Jong's test functions, i.e., f_1 - f_5 which are maximization problems, f_6 - f_8 are converted into equivalent maximization problems by multiplying the achieved fitness value with -1 [6].

3.2 Experimental Setup

Since all the decision variables in De Jong's test functions are continuous-valued, they need to be converted to binary-valued variables before being employed by the BinPSO algorithms. In this study, a binary encoding scheme that is commonly used to perform the conversion in genetic algorithms (GAs) is used. In this scheme, a user-defined *resolution factor* (RF), which is the smallest continuous value represented by a bit, is used to determine the number of bits required to represent a continuous-valued decision variable of a given optimization problem. Given an RF, the number of bits per continuous-valued decision variable, N_b can be obtained as:

$$N_b = \left\lceil \log_2\left(\frac{x_{\max} - x_{\min}}{\mathrm{RF}}\right)\right\rceil \tag{14}$$

where the $x_{\max}$ and $x_{\min}$ denote the maximum and minimum boundary values of the given search space, respectively. After finding N_b, the conversion between continuous-valued and binary-valued decision variables can be performed using:

$$x_b = \frac{x_c - x_{\min}}{\mathrm{RF}} \tag{15}$$

where x_c is the value of the continuous-valued decision variable and x_b is the decimal value of the binary equivalent of x_c. Then, x_b is converted to its binary equivalent in the form of bitstring consisting of N_b bits. After that, all the bitstrings are concatenated to a single bitstring, forming a solution vector which consists of $n \times N_b$ binary-valued decision variables. In this study, the RFs chosen for each test function is given in Table 1.

As the CBinPSO algorithm contains multiple swarms while other BinPSO variants contain only one swarm, the number of iterations is not accurate as a processing time measure. This is because, in one iteration, the CBinPSO algorithm spends a number of function evaluations (FEs) equivalent to K swarms multiplied by the number of particles of one swarm whereas other BinPSO variants spend only a number of FEs equivalent to the number of particles of one swarm in one iteration. Thus, the number of FEs is used as the processing time measure for a fair complexity comparison. The total number of FEs is given as $FE = N_t N_s K$ for the CBinPSO algorithm and $FE = N_t N_s$ for other BinPSO variants, where N_t is the number of iterations.

Table 1. Parameter settings of test functions and their global maxima

F	Function Name	n	RF	N_b	n_b	Domain $[x_{\min}, x_{\max}]^n$	Global Maxima
f_1	Sphere	3	0.01	10	30	$[-5.12, 5.12]^n$	78.64
f_2	Rosenbrock	2	0.001	12	24	$[-2.048, 2.048]^n$	3905.93
f_3	Step	5	0.01	10	50	$[-5.12, 5.12]^n$	55.0
f_4	Noisy Quadric	30	0.01	8	240	$[-1.28, 1.28]^n$	1248.2
f_5	Foxholes	2	0.001	17	34	$[-65.536, 65.536]^n$	500.0
f_6	Rastrigin	30	0.01	10	300	$[-5.12, 5.12]^n$	0
f_7	Ackley	30	0.1	10	300	$[-30, 30]^n$	0
f_8	Griewank	30	0.1	13	390	$[-300, 300]^n$	0

For f_1 - f_5, the simulation parameters are set as follows: N_s, FE and the number of simulation repetitions are set to 20, 4000 and 20, respectively. For f_6 - f_8, N_s, FE and the number of simulation repetitions are set to 40, 40000 and 30, respectively. The mutation rate of the MBinPSO algorithm is set to 0.3, 0.7, 0.5, 0.7, 0.7, 0.0, 0.0 and 0.4 for for $f_1, f_2, f_3, f_4, f_5, f_6, f_7$ and f_8, respectively as in [12]. Additionally, K is set to n of each test function for the CBinPSO algorithm (see Table 1). The rest of the parameters of the BinPSO, NBinPSO, EPSO and EPSOq, and MBinPSO algorithms are set as in [7], [11], [2] and [12], respectively.

3.3 Results and Discussion

Table 2 tabulates the mean and standard deviation of the objective function values achieved by each BinPSO variant on the test functions where the values in bold indicate the best results. Overall, the CBinPSO algorithm attains the best solution quality as it optimizes most of the test functions. The NBinPSO, EPSO and EPSOq algorithms can only obtain optimal solutions for f_1, f_2, f_4 and f_5. Other BinPSO variants do not perform well on all the test functions. As such, the CBinPSO algorithm is generally the best algorithm in terms of solution quality for both unimodal and multimodal optimization problems. On the other hand, the NBinPSO, EPSO and EPSOq algorithms is only suitable for unimodal and simple multimodal problems.

The performance of the EPSOq algorithm (with GBest) on De Jong's test functions is generally more stable compared to other BinPSO variants, as shown in Table 2. Its stability on f_3 is outperformed by only the CBinPSO algorithm. On the other hand, the CBinPSO algorithm demonstrates more stable performance on f_6 and f_8, compared to other BinPSO variants. Additionally, as the CBinPSO algorithm is the only one that finds near-optimal solutions to the multimodal problems, i.e., f_6 - f_8, its comparison against other BinPSO variants in terms of performance stability is thus trivial.

Fig. 3(a)-(e) shows the convergence performance of the BinPSO variants in maximizing the De Jong's test functions. The NBinPSO, EPSO and EPSOq algorithms converge within 300 FEs to the optimal solutions for f_1, f_2, f_4 and f_5, which are the fastest among all the other BinPSO variants. Although these BinPSO variants also converge within 300 FEs for f_3, they converge to a suboptimal solution. This implies that the BinPSO variants are vulnerable to the local optimums of f_3 and likely to be trapped in these values. The CBinPSO algorithm converges slower than the NBinPSO, EPSO and EPSOq algorithms for f_1 - f_5. This is because the CBinPSO algorithm spends more FEs in each iteration. In particular, the CBinPSO algorithm is unable to reach convergence for f_4 because the number of FEs given is insufficient to converge. The BinPSO and MBinPSO algorithms converge to local optimums of all the test functions, thus it is trivial to compare them with other BinPSO variants in terms of convergence speed. In summary, the NBinPSO, EPSO and EPSOq are more time-efficient for solving unimodal and simple multimodal problems.

Table 2. Mean and standard deviation of the achievable fitness scores

PSOs	f_1	f_2	f_3	f_4
BinPSO–Gbest	$77.94\pm8.767\times10^{-1}$	$3749\pm1.116\times10^{2}$	$53.30\pm8.645\times10^{-1}$	$735.5\pm5.623\times10^{1}$
BinPSO–Lbest	$78.12\pm4.897\times10^{-1}$	$3529\pm3.986\times10^{2}$	$53.55\pm1.234\times10^{0}$	$792.6\pm6.779\times10^{1}$
NBinPSO	$\mathbf{78.64\pm1.458\times10^{-14}}$	$\mathbf{3905\pm4.665\times10^{-13}}$	$44.65\pm2.277\times10^{0}$	$\mathbf{1248}\pm2.956\times10^{-1}$
EPSO–Gbest	$\mathbf{78.64\pm1.458\times10^{-14}}$	$\mathbf{3905\pm4.665\times10^{-13}}$	$41.50\pm2.856\times10^{0}$	$\mathbf{1248}\pm2.938\times10^{-1}$
EPSO–Lbest	$\mathbf{78.64\pm1.458\times10^{-14}}$	$\mathbf{3905\pm4.665\times10^{-13}}$	$41.10\pm2.673\times10^{0}$	$\mathbf{1248}\pm2.939\times10^{-1}$
EPSOq–Gbest	$\mathbf{78.64\pm1.458\times10^{-14}}$	$\mathbf{3905\pm4.665\times10^{-13}}$	$42.00\pm2.733\times10^{0}$	$\mathbf{1248\pm2.850\times10^{-1}}$
EPSOq–Lbest	$\mathbf{78.64\pm1.458\times10^{-14}}$	$\mathbf{3905\pm4.665\times10^{-13}}$	$42.10\pm2.593\times10^{0}$	$\mathbf{1248}\pm2.887\times10^{-1}$
MBinPSO–Gbest	$74.75\pm1.553\times10^{0}$	$3733\pm8.859\times10^{1}$	$48.00\pm7.255\times10^{-1}$	$554.0\pm2.813\times10^{1}$
MBinPSO–Lbest	$74.02\pm1.851\times10^{0}$	$3736\pm8.318\times10^{1}$	$48.60\pm1.231\times10^{0}$	$543.4\pm2.067\times10^{1}$
CBinPSO	$\mathbf{78.64\pm1.458\times10^{-14}}$	$3901\pm4.181\times10^{0}$	$\mathbf{55.00\pm0.000\times10^{0}}$	$\mathbf{1248}\pm1.664\times10^{1}$
PSOs	f_5	f_6	f_7	f_8
BinPSO–Gbest	$\mathbf{499.9}\pm3.559\times10^{-6}$	$-349.8\pm2.881\times10^{1}$	$\mathbf{-20.17\pm6.100\times10^{-2}}$	$-151.5\pm1.640\times10^{1}$
BinPSO–Lbest	$\mathbf{499.9}\pm1.416\times10^{-6}$	$-384.4\pm3.063\times10^{1}$	$\mathbf{-20.14\pm4.670\times10^{-2}}$	$-229.3\pm1.767\times10^{1}$
NBinPSO	$\mathbf{499.9\pm2.916\times10^{-13}}$	$-400.4\pm2.636\times10^{1}$	$\mathbf{-20.62\pm8.200\times10^{-2}}$	$-225.3\pm2.392\times10^{1}$
EPSO–Gbest	$\mathbf{499.9\pm2.916\times10^{-13}}$	$-423.2\pm2.013\times10^{1}$	$-19.99\pm1.475\times10^{-1}$	$-231.2\pm2.682\times10^{1}$
EPSO–Lbest	$\mathbf{499.9\pm2.916\times10^{-13}}$	$-404.1\pm2.683\times10^{1}$	$-19.94\pm2.703\times10^{-4}$	$-231.2\pm2.134\times10^{1}$
EPSOq–Gbest	$\mathbf{499.9\pm2.916\times10^{-13}}$	$-410.7\pm2.275\times10^{1}$	$-20.16\pm2.778\times10^{-1}$	$-233.5\pm1.716\times10^{1}$
EPSOq–Lbest	$\mathbf{499.9\pm2.916\times10^{-13}}$	$-409.7\pm3.073\times10^{1}$	$-19.94\pm7.754\times10^{-4}$	$-224.5\pm2.336\times10^{1}$
MBinPSO–Gbest	$\mathbf{499.9}\pm6.199\times10^{-6}$	$-43.88\pm7.944\times10^{0}$	$-15.45\pm3.918\times10^{0}$	$-149.9\pm1.044\times10^{1}$
MBinPSO–Lbest	$\mathbf{499.9}\pm6.783\times10^{-6}$	$-378.9\pm1.947\times10^{1}$	$\mathbf{-20.06\pm1.698\times10^{-1}}$	$-147.5\pm1.198\times10^{1}$
CBinPSO	$\mathbf{499.9}\pm3.604\times10^{-13}$	$\mathbf{-0.668\pm2.803\times10^{-1}}$	$\mathbf{-0.875\pm3.760\times10^{-2}}$	$\mathbf{-0.164\pm1.831\times10^{-1}}$

For f_6 - f_8, the NBinPSO, EPSO and EPSOq algorithms prematurely converge within 5000 FEs to solutions at which the achievable fitness scores are significantly lower than their global maximum, as shown in Fig. 3(f)-(g). This indicates that these BinPSO variants suffer from premature convergence for complex multimodal problems and hence they are not suitable for such problems. Though the BinPSO and MBinPSO algorithms achieve better solutions for the complex multimodal functions, they need more FEs to reach convergence. The CBinPSO algorithm converges to near-optimal solutions after around 15000 FEs for f_6 and f_7 and after 5000 FEs for f_8, which is reasonably fast. This shows that the CBinPSO algorithm is more time-efficient for solving complex multimodal functions.

3.4 Real-World Engineering Problem

In this section, the BinPSO variants are tested on a real-world application, which is the cooperative spectrum sensing problem in a cognitive radio network. This problem is to maximize the probability of detecting the occupancy of a primary wireless channel by a group of secondary or cognitive radio users as in [4]. For this problem, the number of cognitive radio users is set to 20, each weighting coefficient is binary-encoded with RF = 0.0001, and the other network parameter settings follow those in [4]. For the CBinPSO algorithm, K is set to 20. For the MBinPSO algorithm, the

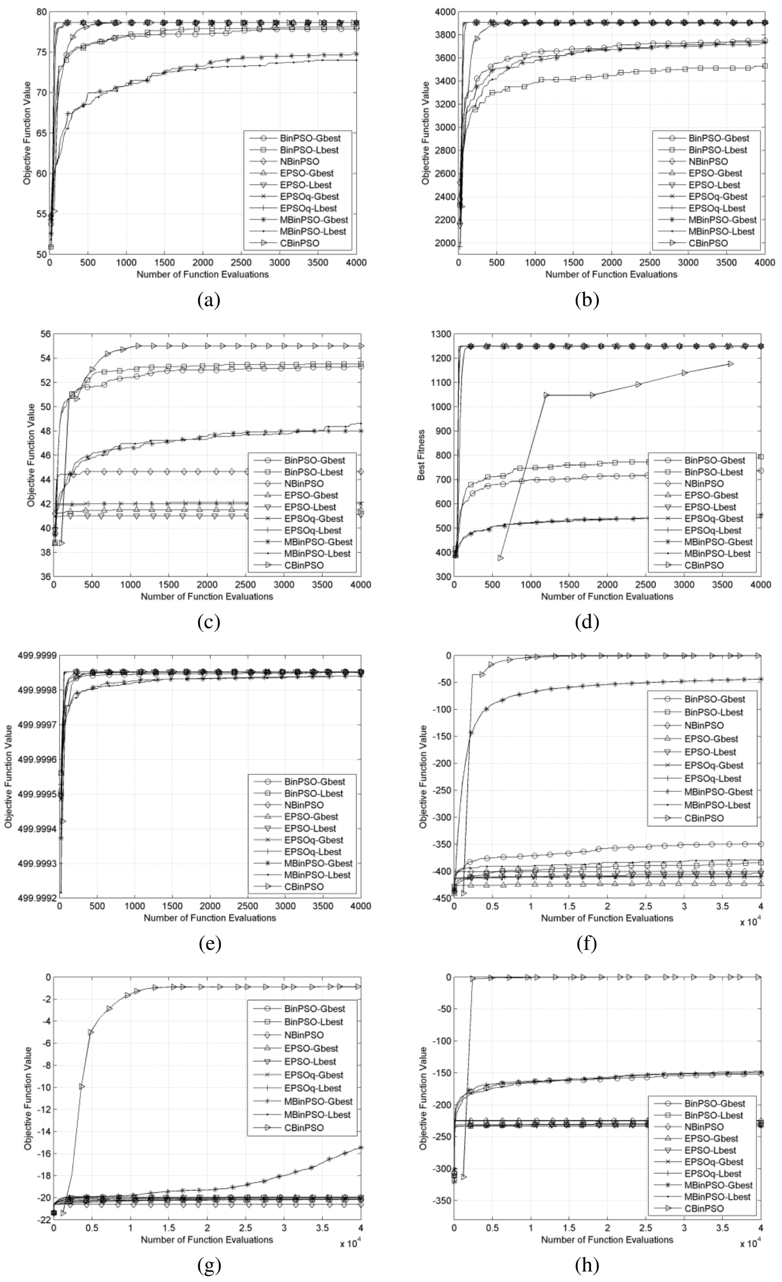

Fig. 3. Convergence performance on (a) f_1, (b) f_2, (c) f_3, (d) f_4, (e) f_5, (f) f_6, (g) f_7 and (h) f_8

mutation rate is set to 0.5. For all the BinPSO variants, N_s, *FE* and the number of simulation repetitions are set to 40, 3000 and 100, respectively. Other BinPSO parameters are the same as in the previous section.

The CBinPSO algorithm is shown to achieve the best solution quality and performance stability, as shown in Table 3. In Fig. 4, the EPSO algorithm attains the highest convergence speed while the CBinPSO and MBinPSO algorithms are the Overall, the CBinPSO algorithm is generally the best performer due to its high solution quality and stability. Although its convergence speed is slow, it almost reaches convergence in the given number of FEs.

Table 3. Mean and standard deviation achieved for the cooperative spectrum sensing problem

PSO	BinPSO		NBin PSO	EPSO		EPSOq		MBinPSO		CBin-PSO
	Gbest	Lbest		Gbest	Lbest	Gbest	Lbest	Gbest	Lbest	
Result	0.591 ± 0.014	0.589 ± 0.014	0.651 ± 0.036	0.751 ± 0.037	0.747 ± 0.038	0.655 ± 0.050	0.655 ± 0.056	0.740 ± 0.037	0.675 ± 0.089	**0.858 ± 0.004**

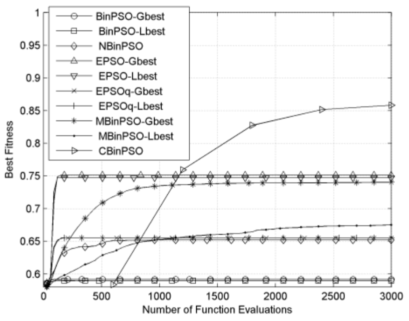

Fig. 4. Convergence performance

4 Conclusion and Future Works

In this paper, we have investigated the performance of various BinPSO variants in terms of solution quality, performance stability and convergence speed. The NBinPSO, EPSO and EPSOq algorithms are more suitable for solving unimodal and simple multimodal problems due to their high solution quality, stability and convergence speed on such problems, compared to other BinPSO variants. Although the CBinPSO algorithm can also achieve high solution quality on the problems, the NBinPSO, EPSO and EPSOq algorithms are more processing time-efficient. On the other hand, the CBinPSO algorithm is more suitable for solving complex multimodal problems as it achieves high solution quality and stability with reasonably fast convergence, compared to other BinPSO variants. The CBinPSO algorithm also performs well on the cooperative spectrum sensing problem in cognitive radio networks. Overall, the CBinPSO algorithm is the most promising candidate for various types of problems.

Nevertheless, improvement on the convergence speed of CBinPSO algorithm is still required when applied to other real-time applications.

References

1. Chatterjee, A., Tudu, B., Paul, K.C.: Towards optimized binary pattern generation for grayscale digital halftoning: A binary particle swarm optimization (BPSO) approach. J. Vis. Commun. Image R. **23**(8), 1245–1259 (2012)
2. Chen, E., Li, J., Liu, X.: In search of essential binary discrete particle swarm. Appl. Soft Comput. **11**(3), 3260–3269 (2011)
3. De Jong, K. A.: An analysis of the behaviour of a class of genetic adaptive systems. Unpublished doctoral dissertation, University of Michigan, Ann Arbor, MI, US (1975)
4. El-Saleh, A.A., Ismail, M., Ali, M.A.M.: Genetic algorithm-assisted soft fusion-based linear cooperative spectrum sensing. IEICE Electron. Express **8**(18), 1527–1533 (2011)
5. Jin, N., Rahmat-Samii, Y.: Advances in particle swarm optimization for antenna designs: real-number, binary, single-objective and multiobjective implementations. IEEE Trans. Antennas Propag. **55**(3), 556–567 (2007)
6. Kameyama, K.: Particle swarm optimization – a survey. IEICE Trans. Inf. & Syst. **E92-D**(7), 1354–1361 (2009)
7. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publisher, San Francisco (2001)
8. Kennedy, J., Eberhart, R. C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, WA (1995)
9. Kennedy J., Eberhart, R. C.: A discrete binary version of the particle swarm algorithm. In: IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL (1997)
10. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Proceedings of the IEEE Congress on Evolutionary Computing, Honolulu, HI, USA (2002)
11. Khanesar, M.A., Teshnehlab, M., Shoorehdeli, M. A.: A novel binary particle swarm optimization. In: Mediterranean Conference on Control and Automation, Athens, Greece (2007)
12. Lee, S., Soak, S., Oh, S., Pedrycz, W., Jeon, M.: Modified binary particle swarm optimization. Prog. Natural Sci. **18**(9), 1161–1166 (2008)
13. Messerschmidt, L., Engelbrecht, A.P.: Learning to play games using a PSO-based competitive learning approach. IEEE Trans. Evol. Comput. **8**(3), 280–288 (2004)
14. Pookpunt, S., Ongsakul, W.: Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. Renew. Energ. **55**, 266–276 (2013)
15. Sarath, K.N.V.D., Ravi, V.: Association rule mining using binary particle swarm optimization. Eng. App. Artif. Intel. **26**(8), 1832–1840 (2013)
16. Schutte, J.F., Groenwold, A.A.: Sizing design of truss structures using particle swarms. Struct. Multidisc. Optim. **25**(4), 261–269 (2003)
17. Tasgetiren, M.F., Liang, Y.: A binary particle swarm optimization algorithm for lot sizing problem. J. Econ. Soc. Res. **5**(2), 1–20 (2003)

18. Van den Bergh, F., Engelbrecht, A.P.: Cooperative learning in neural networks using particle swarm optimizers. South African Comput. J. **26**, 84–90 (2000)
19. Van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 225–239 (2004)
20. Yuan, X., Nie, H., Su, A., Wang, L., Yuan, Y.: An improved binary particle swarm optimization for unit commitment problem. Expert Syst. Appl. **36**(4), 8049–8055 (2009)
21. Zhao, Z., Xu, S., Zheng, S., Shang, J.: Cognitive radio adaptation using particle swarm optimization. Wirel. Commun. Mob. Comput. **9**(7), 875–881 (2009)