

# Optimal Power Allocation of Wireless Sensor Networks with Multi-operator Based Constrained Differential Evolution

Yanan Li, Wenyin Gong<sup>(✉)</sup>, and Zhihua Cai

School of Computer Science, China University of Geosciences,  
Wuhan 430074, People's Republic of China  
wygong@cug.edu.cn

**Abstract.** Optimal power allocation (OPA) is considered to be one of the key issues in designing a wireless sensor network (WSN). Generally, the OPA in WSN can be formulated as a numerical optimization problem with constraints. Differential evolution (DE) is a powerful evolutionary algorithm for numerical, however, the success of DE in solving a specific problem crucially depends on appropriately choosing suitable mutation strategy and its associated control parameter values. Meanwhile, there is no single parameter setting and strategy that is able to consistently obtain the best results for the OPA with different number of sensor nodes. Based on the above considerations, in this paper, a multi-operator based constrained differential evolution is proposed, where *probability matching* and *constrained credit assignment* techniques are used so as to adaptively select the most suitable strategy in different phase of the search process for the OPA. Additionally, the parameter adaptation technique is used to avoid the fine-tuning of DE parameters for different problems. The proposed algorithm has been evaluated in several OPA with different number of sensor nodes, and its performance is compared with single-strategy based DE variants and other methods. Experimental results indicate that the proposed algorithm is able to provide better results than the compared methods.

## 1 Introduction

A wireless sensor network (WSN) is a network of distributed autonomous devices that can sense or monitor physical or environmental conditions cooperatively. In WSNs, there exists a large number of small, inexpensive, spatially distributed sensor nodes that are deployed in an ad hoc manner in vast geographical areas for remote operations and these nodes can acquire, process, and transmit data over wireless medium. WSNs also are used in numerous applications such as security detection, traffic tracking, environmental monitoring, because of characteristics of the low cost and ease of operation [1]. In addition, battery supply, storage resources and commutation bandwidth tremendously restrict the communication and computational capabilities of sensor nodes [2]. Therefore, there has been significant research concentrate that revolves around reaping and minimizing

energy. But above all, it may be impossible to replace or change the batteries of the sensor nodes because of cost and operating environment considerations [3]. Therefore, the optimal power allocation can be considered as one of the crucial issues for designing a WSN.

Differential evolution (DE), which was firstly proposed by Storn and Price [4], is one of the most powerful EAs for global numerical optimization. The advantages of DE are its simple structure, ease of use, speed, and robustness, which leads it on many applications, such as data mining, neural network training, pattern recognition, digital filter design, engineering design, etc. [5]. More applications of DE can be found in the literature [6].

The OPA in WSN can be seen as a constrained numerical optimization problem [3]. In addition, the optimal power allocation in WSNs have attained considerable attention. Recently, there are several algorithms are proposed for OPA, such as particle swarm optimization (PSO) [3, 7, 8], hybrid DE with biogeography-based optimization [9]. Both the two algorithms achieved good performance for independent and identically distributed (i.i.d.) and correlated data fusion in WSNs. In addition, due to the advantages of DE in the numerical optimization, recently, DE has been successfully used for the constrained optimization problems (COPs) by means of employing the constraint-handling techniques [10]. Meanwhile there is no single parameter setting and strategy that is able to consistently obtain the best results for the OPA with different number of sensor nodes [11]. Therefore, it is worth exploring more effective algorithm for OPA.

Differential mutation is the crucial operator in DE which is operated based on the distribution of solutions in the current population. New offsprings are created by combining the parent individual and the mutant individual. Only if the offspring has better fitness value, it can replace its parent. However, it is a difficult and crucial task to choose which mutation strategy for the performance of the DE [12, 13]. Based on the above consideration, in this paper, we proposed a multi-operator differential evolution based on *Probability Matching* and *constrained credit assignment* for solving the OPA. The proposed method is referred to as PM-MDE, in short.

## 2 Preliminaries

In this section, we first briefly describe the optimal power allocation in WSNs, followed by the description of the classical DE algorithm. Then, the constraint-handling technique used in this work is introduced. Finally, adaptive strategy selection is presented.

### 2.1 Optimal Power Allocation in WSNs

Generally, the power allocation problem can be considered as a constrained numerical optimization problem and it also can be formulated as follows [3]:

$$\min_{G_k \geq 0} \sum_{k=1}^L G_k^2 \quad (1)$$

subject to

$$\begin{aligned}
 P(E) &= Q\left(\frac{1}{2}\sqrt{m^2 e^T A \sum_L^{-1} A e}\right) \leq \varepsilon, \\
 G_k &\geq 0, \\
 K &= 1, \dots, L.
 \end{aligned}
 \tag{2}$$

where  $\varepsilon$  is the required fusion error probability threshold,  $L$  indicates the number of sensor nodes,  $m$  represents the deterministic signal and  $G_K$  is the amplifier gain at node  $k$ .  $e$  is the  $L$ -length vector with all ones. The covariance matrix is  $\sum_L = A^T \sum_v A + \sum_w$ , where  $A = \text{diag}(H_1 G_1, \dots, H_L G_L)$ ,  $\sum_v$  is the observation and  $\sum_w$  is receiver noise covariances. Specially, when the local observations and the receiver noise are both i.i.d., the probability of fusion error can be simplified to:

$$P(E) = Q\left(\frac{m}{2}\sqrt{\sum_{k=1}^L \frac{H_k^2 G_k^2}{\delta_v^2 H_k^2 G_k^2 + \delta_w^2}}\right) \leq \varepsilon.
 \tag{3}$$

The inequality in (3) can be expressed as follows:

$$\beta \leq \sqrt{\sum_{k=1}^L \frac{H_k^2 G_k^2}{\delta_v^2 H_k^2 G_k^2 + \delta_w^2}}
 \tag{4}$$

where  $\beta = \frac{2}{m} Q^{-1}(\varepsilon)$  and  $Q(\cdot)$  is the complementary Gaussian cumulative distribution function.  $\delta_v$  means the variances of the observation noise and  $\delta_w$  represents the receiver noise.  $H_k$  indicates the channel fading coefficient. It is needed to point that  $H_k$  follows an exponential distribution (i.e., Rayleigh fading) with unit mean [9].

Besides the above situation, when the sensor observations are spatially correlated, the observation noise covariance matrix  $\sum_v$  can be formulated as follows [14]:

$$\sum_v = \delta_v^2 \begin{pmatrix} 1 & \rho^d & \dots & \rho^{d(L-2)} & \rho^{d(L-1)} \\ \rho^d & 1 & \dots & \rho^{d(L-3)} & \rho^{d(L-2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho^{d(L-1)} & \rho^{d(L-2)} & \dots & \rho^d & 1 \end{pmatrix}
 \tag{5}$$

The inequality in (3) can be expressed as follows:

$$\beta \leq \sqrt{e^T A \sum_n^{-1} A e}
 \tag{6}$$

where  $d_j = d(j-1)$ ,  $j = 1, \dots, L$ , which means the sensor nodes are equally spaced along a straight line. Because  $\sum_v$  is not diagonal, it is difficult to evaluate  $\sum_n^{-1}$  in closed form. Therefore, it is necessary to introduce a specific sensor network model which derives an upper bound for  $P(E)$  and is proposed by Wimalajeewa and Jayaweera [3]. Finally, DE algorithm becomes suitable to use for OPA under arbitrary correlated observation.

## 2.2 Differential Evolution

DE is a simple yet efficient evolutionary algorithm (EA) for global numerical optimization [4]. For the population initialization, a uniform distribution is usually used in the literature within the search space. After initialization, DE generates offsprings based on combining the parent individual and several other individuals of the same population which means that offsprings are generated according to mutation operation and crossover operation. An offspring is evaluated by a fitness function and then replaces the parent individual only if it has an equal or better fitness value. DE repeats this procedure that generates offsprings and replaces parent individual until a predefined termination criterion is satisfied. Generally, the terminal conditions can be fixed either the maximum number of fitness function evaluations (*Max\_NFFEs*) or define a desired solution value to be reached (*VTR*).

## 3 Multi-operator Based Differential Evolution (PM-MDE)

In this section, we will introduce our proposed PM-MDE algorithm in detail. It is previously mentioned that there are many mutation strategies in DE and it is hard to choose the most suitable strategy for different problem in the different stage of evolution. Therefore, it is significant to autonomously select appropriate mutation strategy for OPA. To achieve this performance, in this work, we propose the multi-operator differential evolution for OPA based on *Probability Matching* (PM) technique and *credit assignment* method [15]. The PM technique and the credit assignment method are integrated into DE to implement the adaptive strategy selection and the relative fitness improvement calculation. Moreover, the parameter adaptation method of *CR* and *F* proposed in [16] is adopted in this work.

### 3.1 Strategy Selection and Probability Matching

Following [17], in this subsection the adaptive strategy selection will be introduced. Firstly, it can be supposed that we have  $K$  ( $K > 1$ ) strategies in the pool  $A = \{a_1, \dots, a_k\}$  and a probability vector can be described like  $P(t) = \{p_1(t), \dots, p_K(t)\}$  ( $\forall t : p_{\min} \leq p_i(t) \leq 1; \sum_{i=1}^K p_i(t) = 1$ ). The  $r_a(t)$  presents reward which be achieved by a strategy  $a$  after its application at time  $t$ . The PM method is used to adaptively update the probability  $P_a(t)$  of each strategy  $a$  based on its reward.  $q_a(t)$  is the known quality (or empirical estimate) of a strategy  $a$ , that is updated as follows [18]:

$$q_a(t+1) = q_a(t) + \alpha \cdot [r_a(t) - q_a(t)], \quad (7)$$

where  $\alpha \in (0, 1]$  is the adaptation rate. The PM method updates the probability  $P_a(t)$  as follows [15, 18]:

$$p_a(t+1) = p_{\min} + (1 - K \cdot P_{\min}) \frac{q_a(t+1)}{\sum_{i=1}^K q_i(t+1)}. \quad (8)$$

where  $p_{\min} \in (0, 1]$  represents the minimum probability value of each strategy, the objective of this is to ensure no operator gets lost. It is indicated by Eq. (8) that when only one strategy achieves a reward during a long period of time and all other strategies receive no reward, then its selection probability  $P_a(t)$  converges to  $p_{\max} = p_{\min} + (1 - K \cdot p_{\min})$ . It also can be seen that  $\sum_{a=1}^K p_a(t) = 1$  and  $0 < P_{\min} < \frac{1}{K}$ .

### 3.2 Constrained Credit Assignment

As mentioned above, the OPA problem is a constrained optimization problem. Therefore, to assign the credit for each search operator, we need to design the constrained credit assignment technique. In this work, the constraint-handling technique named improved adaptive trade-off model [19] is employed, which was proposed by Wang and Cai and is the improved version of ATM [20]. In IATM, the population can be divided into three situations such as infeasible situation, semi-feasible situation and feasible situation.

In the infeasible situation, the solutions are measured based on their constraint violation. In the semi-feasible situation, the population includes both feasible and infeasible solutions and it can be divided into the feasible group ( $Z_1$ ) and the infeasible group ( $Z_2$ ) based on the feasibility of each solution. The infeasible individuals in the semi-feasible situation should be handled and evaluated based on their constraint violation. Then, the objective function can be seen as  $f(\mathbf{x})$  and  $\mathbf{x}$  is the vector of the solution. The objective function value  $f(\mathbf{x}_i)$  of the solution  $\mathbf{x}_i$  is converted into

$$f'(\mathbf{x}_i) = \begin{cases} f(\mathbf{x}_i), & i \in Z_1 \\ \max\{\varphi \cdot f(\mathbf{x}_{best}) + (1 - \varphi) \cdot f(\mathbf{x}_{worst}), f(\mathbf{x}_i)\}, & i \in Z_2 \end{cases} \quad (9)$$

where  $\varphi$  represents the feasibility ratio of the last population, and  $\mathbf{x}_{best}$  and  $\mathbf{x}_{worst}$  are the best and worst individual in the feasible group  $Z_1$ , respectively. After achieving the changed objective function value of each individual, then it should be normalized as

$$f_{nor}(\mathbf{x}_i) = \frac{f'(\mathbf{x}_i) - \min_{j \in Z_1 \cup Z_2} f'(\mathbf{x}_j)}{\max_{j \in Z_1 \cup Z_2} f'(\mathbf{x}_j) - \min_{j \in Z_1 \cup Z_2} f'(\mathbf{x}_j)}. \quad (10)$$

In addition, the normalized constraint violation can be evaluated as

$$C_{nor}(\mathbf{x}_i) = \begin{cases} 0, & i \in Z_1 \\ \frac{C(\mathbf{x}_i) - \min_{j \in Z_2} C(\mathbf{x}_j)}{\max_{j \in Z_2} C(\mathbf{x}_j) - \min_{j \in Z_2} C(\mathbf{x}_j)}, & i \in Z_2 \end{cases} \quad (11)$$

where  $C(\mathbf{x})$  represents the distance of the solution  $\mathbf{x}$  from the boundaries of the feasible set, which also reflects the degree of its constraint violation.

Then, the final fitness function is obtained as follows:

$$f_{final}(\mathbf{x}_i) = f_{nor}(\mathbf{x}_i) + C_{nor}(\mathbf{x}_i). \quad (12)$$

In the feasible situation, it is to say that all solutions are feasible in the population, the performance of the situation can be evaluated by the objective function value.

Based on the above fitness transformation technique, similar to the method in [21], the relative fitness improvement  $\eta_i$  can be calculated by

$$\eta_i = \begin{cases} \frac{\delta_1}{f(\mathbf{u}_i)} \cdot (f(\mathbf{x}_i) - f(\mathbf{u}_i)), & \text{feasible situation} \\ \frac{\delta_2}{C(\mathbf{u}_i)} \cdot (C(\mathbf{x}_i) - C(\mathbf{u}_i)), & \text{infeasible situation} \\ \frac{\delta_3}{f_{\text{final}}(\mathbf{u}_i)} \cdot (f_{\text{final}}(\mathbf{x}_i) - f_{\text{final}}(\mathbf{u}_i)), & \text{semi-feasible situation} \end{cases} \quad (13)$$

where  $i = 1, \dots, \mu$ .  $\delta$  is the objective fitness of the best-so-far solution in the population.  $\mathbf{x}_i$  and  $\mathbf{v}_i$  are the parent and its offspring, respectively.

In [17], four different credit assignment methods are presented, and the averaged normalization reward is able to provide highly-competitive results through benchmark functions. Based on this consideration, the averaged normalization reward is selected for the credit assignment in this work and is listed as follows:

$$r_a(t) = \frac{r'_a(t)}{\max_{a=1, \dots, K} r'_a(t)} \quad (14)$$

where  $r'_a(t)$  is calculated as

$$r'_a(t) = \frac{\sum_{i=1}^{|S_a|} |S_a|}{|S_a|} \quad (15)$$

and  $S_a$  is the set of all relative fitness improvement  $\eta_i$  of a strategy  $a$  ( $a = 1, \dots, K$ ) at generation  $t$ .

### 3.3 Strategy Pool

DE has realized using different mutation strategies to achieve different performance for solving different problems. Instead of employing the computationally enormous trial-and-error search for the most suitable mutation strategy, we maintain a strategy candidate pool including four mutation strategies. In this work, we choose several effective mutation strategies commonly referred to in DE literatures and choose some of them to construct the strategy candidate pool which are listed as follows:

(1) “DE/rand/1”:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (16)$$

(2) “DE/current-to-best/1”:

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{best} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (17)$$

(3) “DE/rand-to-best/1”:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{best} - \mathbf{x}_{r1}) + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (18)$$

---

**Algorithm 1.** The pseudo-code of PM-MDE

---

```

1: Set  $\mu_F = 0.5; \mu_{CR} = 0.5$ 
2: Generate the initial population randomly
3: Evaluate the fitness for each individual
4: Set  $K = 4, p_{min} = 0.05, \alpha = 0.3,$  and  $\beta = 0.8$ 
5: For each strategy  $a$ , set  $q_a(t) = 0$  and  $p_a(t) = 1/K$ 
6: while The halting criterion is not satisfied do
7:   Set  $S_{CR} = \emptyset; S_F = \emptyset$ 
8:   for  $i = 1$  to  $NP$  do
9:      $CR_i = rndn_i(\mu_{CR}, 0.1)$   $F_i = rndc_i(\mu_F, 0.1)$ 
10:    Select the strategy  $SI_i$  based on its probability
11:    Select uniform randomly  $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ 
12:     $j_{rand} = rndint(1, D)$ 
13:    for  $j = 1$  to  $D$  do
14:      if  $rndreal_j(0, 1) < CR$  or  $j == j_{rand}$  then
15:        if  $SI_i == 1$  then
16:           $u_{i,j}$  is generated by strategy (16)
17:        else if  $SI_i == 2$  then
18:           $u_{i,j}$  is generated by strategy (17)
19:        else if  $SI_i == 3$  then
20:           $u_{i,j}$  is generated by strategy (18)
21:        else if  $SI_i == 4$  then
22:           $u_{i,j}$  is generated by strategy (19)
23:        end if
24:      else
25:         $u_{i,j} = x_{i,j}$ 
26:      end if
27:    end for
28:  end for
29:  for  $i = 1$  to  $NP$  do
30:    Evaluate the offspring  $\mathbf{u}_i$  based on constraint-handling technique
31:    if  $f(\mathbf{u}_i)$  is better than or equal to  $f(\mathbf{x}_i)$  then
32:       $CR_i \rightarrow S_{CR}; F_i \rightarrow S_F$ 
33:      Replace  $\mathbf{x}_i$  with  $\mathbf{u}_i$ 
34:    end if
35:  end for
36:  Update the value of  $\mu_{CR}$  and  $\mu_F$ 
37:  Calculate the reward  $r_a(t)$  for each strategy
38:  Update the probability  $p_a(t)$  for each strategy
39: end while

```

---

(4) “DE/current-to-rand/1 ”:

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{19}$$

where  $\mathbf{x}_{best}$  is the best individual in the current generation,  $r_1, r_2, r_3, r_4, r_5 \in \{1, \dots, NP\}$  and  $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ .

It is noteworthy that there are many other strategies could also be incorporated in the pool; the above strategies are just used as an example for the evaluation of the proposed method. It must also be pointed out that the size of the strategy pool and the selection of the strategies used in the pool is not sure and there are no theoretical studies as of today on the choice of the optimal number of available strategies and on the selection of strategies to form the strategy pool [12]. Therefore, by combining the above-mentioned two aspects with the DE algorithm, the PM-MDE method is developed. The pseudo-code of PM-MDE is illustrated in Algorithm 1. During evolution, at each generation  $t$ , there is only one strategy  $SI_i$  been selected based on the choice probability of the strategy for each target parent  $i$ . Then the offspring is generated by employing the selected strategy. The relative fitness improvement  $\eta_i$  based on constraint handling technique is calculated and stored in the set  $S_{SI_i}$  after evaluating the offspring. Finally, parameters value of PM-MDE algorithm such as the reward, quality and probability of each strategy are updated. In addition, to remedy the parameter fine-tuning, in this work, the parameter adaptation technique proposed in [16] is used in PM-MDE.

## 4 Experimental Results and Analysis

In this section, we perform comprehensive experiments to evaluate the performance of PM-MDE and compare the results of our methods with a algorithm named CBBO-DE, which is a hybridization of BBO algorithm and DE algorithm for OPA.

### 4.1 Experimental Settings

Without loss of generality, for all experiments, we use the following parameters unless a change is mentioned:

- Population size:  $NP = 100$ ;
- $\mu_{CR} = 0.5$  and  $\mu_F = 0.5$  [4];
- Number of strategies:  $K = 4$ ; minimal probability:  $p_{min} = 0.05$ ; adaptation rate:  $\alpha = 0.3$ ;
- Maximum Number of Fitness Function Evaluations (Max\_NFFE<sub>s</sub>): 30,000.

Moreover, all experiments were run 30 times. Simulations have been carried out for various values of parameters: fusion error threshold  $\varepsilon$ , correlation degree  $\rho$ , and number of sensors ( $L$ ), and the performances of the different algorithms are shown for different combinations:  $\rho = \{0, 0.01, 0.1, 0.5\}$ .  $\rho = 0$  represents the uncorrelated case. The fusion error threshold  $\varepsilon$  takes its values in  $\{0.1, 0.05, 0.01, 0.001\}$ . The observation signal-to-noise ratio (SNR)  $r_0$  was set at 10 dB.



### 4.2 Numerical Results

In this section, in order to compare the performance of the adaptive strategy selection for OPA, the DE with fixed strategy and recently proposed algorithm for OPA are considered. For fair comparison, the adaptive parameters control keeps the same in all simulation. The statistical features (mean, and standard deviation values) of the best feasible solutions obtained after 30 independent runs for each case study are used to evaluate the performance of the competing algorithms. Numerical results of competing algorithms for OPA problems in WSNs are shown in Tables 1, 2, 3, 4 and 5 when the observations are i.i.d and

**Table 1.** Numerical results of DE with fixed strategy and PM-MDE to optimal power allocation in WSNs when the observation are i.i.d. with  $\rho = 0, \varepsilon = 0.1$  and different number of sensors.

$L$	PM-MDE	$DE^1$	$DE^2$	$DE^3$	$DE^4$
	Mean $\pm$ (Std)	Mean $\pm$ (Std)	Mean $\pm$ (Std)	Mean $\pm$ (Std)	Mean $\pm$ (Std)
10	<b>3.17E+00</b> $\pm$ <b>(3.43E-11)</b>	3.17E+00 $\pm$ (3.38E-06)	3.17E+00 $\pm$ (8.65E-09)	3.17E+00 $\pm$ (8.53E-10)	3.17E+00 $\pm$ (1.99E-03)
20	<b>1.93E+00</b> $\pm$ <b>(3.71E-07)</b>	1.93E+00 $\pm$ (1.10E-06)	1.93E+00 $\pm$ (8.52E-06)	1.93E+00 $\pm$ (1.69E-04)	1.97E+00 $\pm$ (1.43E-02)
50	8.67E-01 $\pm$ (4.23E-03)	8.68E-01 $\pm$ (1.24E-03)	<b>8.66E-01</b> $\pm$ <b>(2.45E-04)</b>	8.75E-01 $\pm$ (3.69E-03)	1.24E+00 $\pm$ (9.33E-02)
100	<b>8.52E-01</b> $\pm$ <b>(2.35E-02)</b>	5.78E+01 $\pm$ (3.77E+01)	6.54E+03 $\pm$ (9.84E+02)	1.53E+03 $\pm$ (1.35E+02)	4.07E+00 $\pm$ (9.53E-01)
150	<b>9.59E-01</b> $\pm$ <b>(7.99E-02)</b>	1.31E+00 $\pm$ (8.52E-02)	2.44E+00 $\pm$ (2.75E-01)	8.74E+03 $\pm$ (3.85E+03)	2.45E+02 $\pm$ (7.61E+01)
200	<b>1.08E+00</b> $\pm$ <b>(3.78E-01)</b>	1.28E+01 $\pm$ (5.12E+01)	3.49E+02 $\pm$ (7.76E+01)	8.88E+04 $\pm$ (2.87E+04)	2.00E+03 $\pm$ (5.06E+02)

1, 2, 3, 4 respectively represent DE algorithm with DE/rand/1, DE/best/1, DE/current-to-best/1, DE/rand-to-best/1 strategy.

**Table 2.** Numerical results of PM-MDE with CBBO-DE to optimal power allocation in WSNs when the observation are i.i.d. with  $\rho = 0, \varepsilon = \{0.1, 0.05, 0.01, 0.001\}$  and different number of sensors.

$\varepsilon$	PM-MDE	CBBO-DE
	Mean $\pm$ (Std)	Mean $\pm$ (Std)
$L = 10$		
0.1	3.1727E+00 $\pm$ (9.2849E-05)	<b>3.1725E+00</b> $\pm$ <b>(1.1803E-06)</b>
0.05	<b>5.9254E+00</b> $\pm$ <b>(5.5930E-08)</b>	5.9723E+00 $\pm$ (4.4272E-05)
0.01	1.5130E+01 $\pm$ (3.5448E-08)	1.1530E+01 $\pm$ (8.0724E-10)
0.001	4.0317E+01 $\pm$ (3.9498E-04)	<b>4.0245E+01</b> $\pm$ <b>(5.5876E-06)</b>
$L = 20$		
0.1	1.9343E+00 $\pm$ (1.3791E-03)	<b>1.9333E+00</b> $\pm$ <b>(1.4882E-03)</b>
0.05	<b>3.6141E+00</b> $\pm$ <b>(1.7754E-03)</b>	3.6413E+00 $\pm$ (1.0540E-03)
0.01	9.1009E+00 $\pm$ (4.0275E-03)	<b>9.0985E+00</b> $\pm$ <b>(6.7927E-04)</b>
0.001	2.1601E+01 $\pm$ (4.2999E-03)	<b>2.1598E+01</b> $\pm$ <b>(6.4187E-04)</b>
$L = 50$		
0.1	<b>1.1192E+00</b> $\pm$ <b>(6.2244E-02)</b>	2.7516E+00 $\pm$ (3.9133E-01)
0.05	<b>1.9591E+00</b> $\pm$ <b>(9.7700E-02)</b>	4.5611E+00 $\pm$ (6.5277E-01)
0.01	<b>4.7101E+00</b> $\pm$ <b>(8.9643E-02)</b>	6.2178E+00 $\pm$ (5.0805E-01)
0.001	<b>1.0306E+01</b> $\pm$ <b>(1.2003E-01)</b>	1.1093E+01 $\pm$ (6.0068E-01)

**Table 3.** Numerical results of PM-MDE with CBBO-DE to optimal power allocation in WSNs when the observation are correlated with  $\rho = \{0.01, 0.1, 0.5\}$ ,  $L = 10$ ,  $\varepsilon = \{0.1, 0.05, 0.01, 0.001\}$ .

$\varepsilon$	PM-MDE	CBBO-DE
	Mean $\pm$ (Std)	Mean $\pm$ (Std)
$\rho = 0.01$		
0.1	3.1834E+00 $\pm$ (1.6035E-04)	<b>3.1833E+00 <math>\pm</math> (3.8557E-04)</b>
0.05	<b>5.9502E+00 <math>\pm</math> (4.3685E-07)</b>	5.9975E+00 $\pm$ (4.0340E-05)
0.01	1.5255E+01 $\pm$ (3.3372E-08)	1.5255E+01 $\pm$ (1.8879E-09)
0.001	<b>4.0980E+01 <math>\pm</math> (3.4452E-04)</b>	4.1046E+01 $\pm$ (4.6323E-04)
$\rho = 0.1$		
0.1	<b>3.2792E+00 <math>\pm</math> (9.9562E-05)</b>	3.2834E+00 $\pm$ (1.1153E-04)
0.05	<b>6.1755E+00 <math>\pm</math> (5.6832E-09)</b>	6.2391E+00 $\pm$ (3.8481E-05)
0.01	<b>1.6489E+01 <math>\pm</math> (4.6071E-06)</b>	1.6562E+01 $\pm$ (3.5623E-09)
0.001	<b>4.8644E+01 <math>\pm</math> (1.1203E-10)</b>	4.9077E+01 $\pm$ (7.8106E-04)
$\rho = 0.5$		
0.1	<b>3.5839E+00 <math>\pm</math> (8.5642E-04)</b>	3.8583E+00 $\pm$ (2.3158E-04)
0.05	<b>6.9964E+00 <math>\pm</math> (3.6859E-04)</b>	8.1361E+00 $\pm$ (8.0422E-05)
0.01	<b>2.2803E+01 <math>\pm</math> (8.8792E-04)</b>	3.4349E+01 $\pm$ (8.0846E-07)
0.001	<b>1.0778E+02 <math>\pm</math> (6.7852E+00)</b>	7.3514E+02 $\pm$ (6.6360E-02)

correlated. In addition, the best result among competing algorithms in Tables 1, 2, 3, 4 and 5 are shown in **boldface**.

Firstly, we choose four strategies which are frequently used in DE literature as the competing algorithm in Table 1. From results in the Table 1 we can see that PM-MDE algorithm we proposed is significantly better than DE with fixed strategies on most of the value of  $L$  (sensors) when the observations are i.i.d with  $\rho = 0$  and  $\varepsilon = 0.1$ . Second, in  $L = 50$ , DE algorithm with DE/best/1 obtains best results and better results were found by PM-MDE among all competing algorithms. Important observations about the convergence rate and stability of different algorithms can be made from the results presented in Table 1 and these results suggest that the overall convergence rate of PM-MDE is the best or second best for OPA in the competing algorithms.

Table 2 shows a comparison of the performances of PM-MDE algorithm and CBBO-DE algorithm, for the different values of  $\varepsilon$  and  $L$ , in the uncorrelated case ( $\rho = 0$ ). Firstly, various simulations of PM-MDE with  $\varepsilon$  chosen from  $\{0.1, 0.05, 0.01, 0.001\}$  shown that PM-MDE algorithm emerged the best candidate result for  $L = 10$  and  $L = 50$  sensors in terms of the best mean results. For the  $L = 20$  sensors case, the CBBO-DE produces the best mean results. Secondly, simulation results also indicate that PM-MDE does function efficiently within the large number of sensors.

**Table 4.** Numerical results of PM-MDE with CBBO-DE to optimal power allocation in WSNs when the observation are correlated with  $\rho = \{0.01, 0.1, 0.5\}$ ,  $L = 20$ ,  $\varepsilon = \{0.1, 0.05, 0.01, 0.001\}$ .

$\varepsilon$	PM-MDE	CBBO-DE
	Mean $\pm$ (Std)	Mean $\pm$ (Std)
$\rho = 0.01$		
0.1	<b>1.9394E+00 <math>\pm</math> (1.7861E-03)</b>	1.9396E+00 $\pm$ (2.1426E-03)
0.05	<b>3.6292E+00 <math>\pm</math> (2.6169E-03)</b>	3.6559E+00 $\pm$ (9.7422E-04)
0.01	9.1634E+01 $\pm$ (3.7387E-03)	<b>9.1607E+00 <math>\pm</math> (8.9830E-04)</b>
0.001	2.1842E+01 $\pm$ (3.1386E-03)	2.1842E+01 $\pm$ (3.9883E-03)
$\rho = 0.1$		
0.1	1.9908E+00 $\pm$ (3.5774E-03)	<b>1.9905E+00 <math>\pm</math> (1.3193E-03)</b>
0.05	<b>3.7594E+00 <math>\pm</math> (3.6727E-03)</b>	3.7958E+00 $\pm$ (1.8885E-03)
0.01	<b>9.7554E+01 <math>\pm</math> (5.4935E-03)</b>	9.7894E+01 $\pm$ (9.1896E-04)
0.001	<b>2.4182E+01 <math>\pm</math> (3.4839E-03)</b>	2.4324E+01 $\pm$ (1.9630E-03)
$\rho = 0.5$		
0.1	<b>2.1879E+00 <math>\pm</math> (7.4038E-03)</b>	2.3026E+00 $\pm$ (2.1919E-03)
0.05	<b>4.3232E+00 <math>\pm</math> (2.1397E-02)</b>	4.8357E+00 $\pm$ (2.7079E-03)
0.01	<b>1.2547E+01 <math>\pm</math> (8.5490E-02)</b>	1.5865E+01 $\pm$ (3.2673E-03)
0.001	<b>3.6247E+01 <math>\pm</math> (9.1529E-02)</b>	6.0685E+01 $\pm$ (6.5080E-02)

Tables 3, 4 and 5 show the results of the comparison with CBBO-DE when the observations are correlated in the case of  $L = 10, 20, 50$  sensors for different values of the fusion error probability  $\varepsilon$  and the degree of correlation  $\rho$ , respectively. In the experiments reported above, the results are shown for  $\rho = \{0, 0.01, 0.1, 0.5\}$  and  $\varepsilon = \{0.1, 0.05, 0.01, 0.001\}$ . It can be seen that in each case, PM-MDE respectively outperforms other competing algorithm in 10, 8, and 12. From Table 5, specifically, we can see that PM-MDE has emerged as the best performer since it obtained the best mean results in the all cases. It is similar to the observation from above experiments, where PM-MDE algorithm does show obvious performance improvement for OPA, especially in the large number of sensors PM-MDE algorithm obtains better performance.

Overall, according to the results shown in Tables 1, 2, 3, 4 and 5 and the above analysis, we can conclude that when the observations are i.i.d and correlated (Tables 3, 4 and 5), the performance improvement for PM-MDE, compared to the other competing algorithm, was better for  $L = 10, 20$ , and 50 sensors. Meanwhile, PM-MDE obtains better results for the larger sensors.

**Table 5.** Numerical results of PM-MDE with CBBO-DE to optimal power allocation in WSNs when the observation are correlated with  $\rho = \{0.01, 0.1, 0.5\}$ ,  $L = 50$  sensors,  $\varepsilon = \{0.1, 0.05, 0.01, 0.001\}$  and different number of sensors.

$\varepsilon$	PM-MDE	CBBO-DE
	Mean $\pm$ (Std)	Mean $\pm$ (Std)
$\rho = 0.01$		
0.1	<b>1.2346E+00 <math>\pm</math> (1.1103E-01)</b>	2.7742E+00 $\pm$ (5.6072E-01)
0.05	<b>2.0543E+00 <math>\pm</math> (8.4439E-02)</b>	4.1691E+00 $\pm$ (3.8860E-01)
0.01	4.8249E+00 $\pm$ (1.3068E-01)	6.0532E+00 $\pm$ 7.9833E-01
0.001	<b>1.0521E+01 <math>\pm</math> (7.3975E-02)</b>	1.0746E+01 $\pm$ (3.3081E-01)
$\rho = 0.1$		
0.1	<b>1.2406E+00 <math>\pm</math> (9.4525E-02)</b>	3.0813E+00 $\pm$ (4.3000E-01)
0.05	<b>2.1356E+00 <math>\pm</math> (8.8744E-02)</b>	4.3770E+00 $\pm$ (6.2806E-01)
0.01	5.1088E+00 $\pm$ (8.1419E-02)	6.6532E+00 $\pm$ (1.0134E+00)
0.001	<b>1.1284E+01 <math>\pm</math> (1.2038E-01)</b>	1.1669E+01 $\pm$ (5.8073E-01)
$\rho = 0.5$		
0.1	<b>1.3432E+00 <math>\pm</math> (8.8691E-02)</b>	2.9672E+00 $\pm$ (4.6985E-01)
0.05	<b>2.4334E+00 <math>\pm</math> (5.5883E-02)</b>	4.6214E+00 $\pm$ (5.7670E-01)
0.01	6.2096E+00 $\pm$ (1.1285E-01)	7.4210E+00 $\pm$ (3.2418E-01)
0.001	<b>1.4824E+01 <math>\pm</math> (1.5434E-01)</b>	1.8449E+01 $\pm$ (3.6772E-02)

## 5 Conclusions

Optimal power allocation (OPA) is considered to be one of the key issues in designing a wireless sensor network (WSN). In this paper, multi-operator differential evolution is proposed for the optimal power allocation in WSNs. Combining with the constraint-handling technique, DE can be used to deal with OPA. However, the DE performance mainly depends on mutation and crossover operators. This new algorithm adaptively chooses the suitable strategy for a specific problem, meanwhile the *probability matching* technique and *credit assignment* method are integrated into DE algorithm. In addition, PM-MDE is compared with DE algorithm using fixed strategy and CBBO-DE algorithm proposed in [11] for OPA. The numerical results indicate that PM-MDE has outperformed the other competing algorithms for several types of simulation case studies, including both independent local observation cases and correlated observation cases. It has also been observed that, PM-MDE algorithm function efficiently within the large number of sensors.

**Acknowledgments.** The work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61573324, 61375066, and 61203307.

## References

1. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. *Comput. Netw.* **52**(12), 2292–2330 (2008)
2. Kulkarni, R., Forster, A., Venayagamoorthy, G.: Computational intelligence in wireless sensor networks: a survey. *IEEE Commun. Surv. Tutor.* **13**(1), 68–96 (2011)
3. Wimalajeewa, T., Jayaweera, S.: Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained PSO. *IEEE Trans. Wirel. Commun.* **7**(9), 3608–3618 (2008)
4. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
5. Das, S., Abraham, A., Konar, A.: Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Syst. Man Cybern. A, Syst. Hum.* **38**(1), 218–237 (2008)
6. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **15**(1), 4–31 (2011)
7. Wimalajeewa, T., Jayaweera, S.K.: PSO for constrained optimization: optimal power scheduling for correlated data fusion in wireless sensor networks. In: *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2007*, pp. 1–5 (2007)
8. Kulkarni, R., Venayagamoorthy, G.: Particle swarm optimization in wireless-sensor networks: a brief survey. *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.* **41**(2), 262–267 (2011)
9. Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M.: Hybridizing biogeography-based optimization with differential evolution for optimal power allocation in wireless sensor networks. *IEEE Trans. Veh. Technol.* **60**(5), 2347–2353 (2011)
10. Coello, C.A.C.: Theoretical, numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput. Methods Appl. Mech. Eng.* **191**(11–12), 1245–1287 (2002)
11. Gong, W., Cai, Z.: An empirical study on differential evolution for optimal power allocation in WSNs. In: *International Conference on Natural Computation*, pp. 635–639 (2012)
12. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009)
13. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: *IEEE congress on evolutionary computation (CEC 2005)*, pp. 1785–1791. IEEE (2005)
14. Dow, M.: Explicit inverses of Toeplitz and associated matrices. *ANZIAM J.* **44**(E), 185–215 (2003)
15. Goldberg, D.E.: Probability matching, the magnitude of reinforcement, and classifier system bidding. *Mach. Learn.* **5**(4), 407–425 (1990)
16. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **13**(5), 945–958 (2009)
17. Gong, W., Fialho, A., Cai, Z.: Adaptive strategy selection in differential evolution. In: Branke, J. (Ed.) *Genetic and Evolutionary Computation Conference (GECCO 2010)*, pp. 409–416. ACM Press (2010)

18. Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: Proceedings of Genetic and Evolutionary Computation Conference GECCO 2005, New York, NY, USA, pp. 1539–1546. ACM (2005)
19. Wang, Y., Cai, Z., Zhou, Y., Zeng, W.: An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* **12**(1), 80–92 (2008)
20. Brest, J., Greiner, S., Boškovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **10**(6), 646–657 (2006)
21. Ong, Y.-S., Keane, A.J.: Meta-Lamarckian learning in memetic algorithms. *IEEE Trans. Evol. Comput.* **8**(2), 99–110 (2004)