

Article

# Applying Two-Stage Differential Evolution for Energy Saving in Optimal Chiller Loading

Chang-Ming Lin <sup>1,\*</sup>, Chun-Yin Wu <sup>2</sup>, Ko-Ying Tseng <sup>3</sup>, Chih-Chiang Ku <sup>2</sup> and Sheng-Fuu Lin <sup>1</sup>

<sup>1</sup> Institute of Electrical and Control Engineering, National Chiao Tung University, No. 1001 University Road, Hsinchu 30010, Taiwan; sflin@mail.nctu.edu.tw

<sup>2</sup> Department of Mechanical Engineering, Tatung University, 40 Zhongshan North Road, 3rd Section Taipei 104, Taiwan; cywu@ttu.edu.tw (C.-Y.W.); afatest@hotmail.com (C.-C.K.)

<sup>3</sup> Energy and Environment Research Laboratories, Industrial Technology Research Institute, Rm.820, Bldg.51, 8F, 195, Sec.4, Chung Hsing Rd., Chutung, Hsinchu 31040, Taiwan; optbilly@gmail.com

\* Correspondence: linbarry@itri.org.tw; Tel.: +886-3-591-6187

Received: 9 January 2019; Accepted: 9 February 2019; Published: 15 February 2019



**Abstract:** In Taiwan, over 45% of the energy in common buildings is used for the air-conditioning system. In particular, the chiller plant consumes about 70% of the energy in air-conditioning system. The electric energy consumption of air-condition system in a clean room of semiconductor factory is about 5–10 times of that in a common building. Consequently, the optimal chiller loading in energy saving of building is a vital issue. This paper develops a new algorithm to solve optimal chiller loading (OCL) problems. The proposed two-stage differential evolution algorithm integrated the advantages of exploration (global search) in the modified binary differential evolution (MBDE) algorithm and exploitation (local search) in the real-valued differential evolution (DE) algorithm for finding the optimal solution of OCL problems. In order to show the performance of the proposed algorithm, comparison with other optimization methods has been done and analyzed. The result shows that the proposed algorithm can obtain similar or better solution in comparison to previous studies. It is a promising approach for the OCL problem.

**Keywords:** energy saving; multi-chiller system; optimal chiller loading; two-stage differential evolution

## 1. Introduction

The air-conditioning systems of large-scale commercial buildings in Taiwan account for about 32%–54% of Taiwan's electrical energy consumption, and chiller plants consume more than 70% of the overall energy consumed by air-conditioning systems [1]. In systems where multiple chillers are operated in parallel (multi-chiller systems), each chiller can operate independently; adjusting the chiller operation schedule to provide the venue with a stable refrigeration ton (RT) load and a flexible maintenance schedule [1] is a common practice in large commercial buildings. Because multi-chiller systems are composed of chillers of varying features or even of various types of chillers, the question of how to adjust appropriate numbers of operating chillers and operational control points so that each chiller operates at optimal efficiency is crucial for saving energy in air-conditioning systems.

In recent years, many articles have discussed the optimal chiller loading (OCL) problem. Chang [2,3] proposed using a branch-and-bound method and a Lagrangian multiplier method to solve OCL problems. In addition to traditional numerical methods, numerous heuristic optimization methods have been used to solve OCL problems. Optimal chiller loading that consumed less energy than that of Chang [2,3] was obtained by genetic algorithms (GAs) [4]. Chang et al. [5] applied an evolutionary strategy (ES) to OCL problems and found a lower power consumption with higher precision for chillers than found previously. Particle swarm optimization (PSO), which uses the

social behavior of entities for evolutionary computing, was applied to OCL problems [6]. PSO was more efficient than binary GA and real-valued GA in solving OCL problems. Differential evolution (DE), which uses the characteristics of intergroup differences for evolution, was also used in OCL problems [7]; according to the literature, its results were superior to that of PSO. Coelho et al. [8] proposed using a differential cuckoo search approach to improve the original cuckoo search approach, so that the performance of a differential cuckoo search for OCL problems could be superior to those of GA, PSO, and DE.

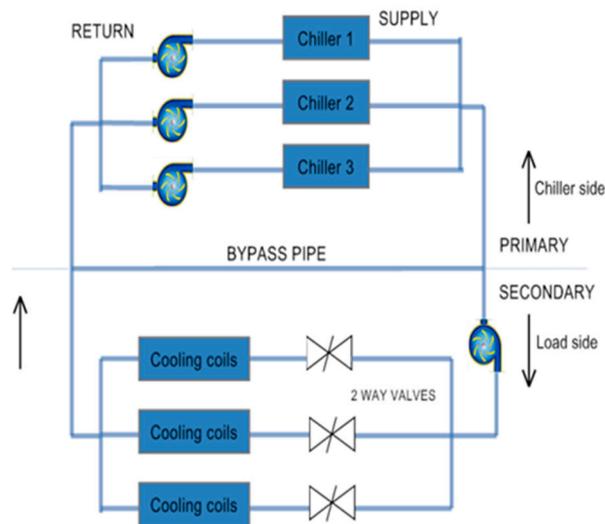
Exploration (global search) and exploitation (local search) are the two critical factors that influence evolutionary optimization methods; the balance of the two directly affects the results and efficacy of searches for optimal solutions. Tan et al. [9] dynamically adjusted evolutionary computations to maintain the balance between scope and convergence of multi-objective optimization. Binkley et al. [10] used PSO for multimodal optimization; when the velocity of the particle swarm was lower than a threshold, the designed quantity was reduced and the particles were restarted to maintain the diversity of the multimodal design space. Epitropakis et al. [11] combined the mutation mechanisms of different DE algorithms, first using exploration and then a high-convergence mechanism to search for optimal solutions. Bao [12] proposed a two-phase hybrid optimization algorithm involving an ant colony algorithm (ACO) and a simulated annealing (SA) optimization algorithm to solve complex optimization problems. Beghi et al. [13] used a multiphase GA for the management of a multi-chiller system to reduce power consumption and operational costs. Cheng and Tran [14] used a two-phase DE algorithm project to obtain an optimized schedule of time and cost.

This study proposes a two-stage DE algorithm to solve OCL problems. The framework includes DE with two different types of variables: the first stage uses a modified binary differential evolution (MBDE) algorithm [15,16] for exploration; the second stage uses a real-valued DE algorithm for exploitation. A binary encoding method enabled greater exploration than a real-valued encoding algorithm [17] would have allowed; thus, in the process of searching for the optimal solution, the proposed method was able to quickly find the optimal solution. After the proposed MBDE completes stage 1, the optimal solution discovered by MBDE in stage 1 undergoes conversion into real numbers and is then introduced into the real-valued DE of stage 2 for optimal exploitation. Through the integration of the aforementioned two stages, the proposed two-stage algorithm, which integrates the advantages of binary MBDE and real-valued DE, has excellent exploration and exploitation capabilities. Its design enhances its operational efficacy and its evolutionary methods enhance the results of its searches for optimal solutions.

The remainder of this paper is structured as follows. Section 2 describes the intricacies of the OCL problem, its objective function, and its restrictions. Section 3 explains the evolutionary mechanism and the features of the two-stage DE algorithm. Section 4 compares the results of the proposed method with those of different methods for various examples. Conclusions are drawn in Section 5.

## 2. Introduction to Multi-chiller System

Multi-chiller systems can provide flexible operation, reserve capacity, and less frequent system shutdowns for maintenance. Those systems composed of two or more chillers are widely used in the air-conditioning systems of large buildings. In a multi-chiller system, each chiller can operate independently and provide different refrigeration capabilities; the chillers operate efficiently according to different or similar performance curves to meet a wide range of RT requirements in HVAC (heating, ventilation and air conditioning) system. The architecture of the multi-chiller system is as shown in Figure 1 below.



**Figure 1.** Multi-chiller system architecture.

Generally, the maximum capacity of a chiller is designed to meet the maximum peak load demand, but because of actual venue requirements and change of seasons, maximum peak load generally only occurs in summer, and the system operates at low partial load mode during the remaining time. Therefore, if the designed capacity is too large, the system consumes excessive power. The partial load rate (PLR) of the chiller can be expressed as (1):

$$PLR = \frac{\text{chiller load}}{\text{chiller rated load}} \quad (1)$$

The power consumption of a chiller and its PLR share a certain relationship; according to the chiller power consumption equation of [8] as shown in (2) and (3):

$$P_i = a_i + b_i \times PLR_i + c_i \times PLR_i^2 \quad (2)$$

$$P_i = a_i + b_i \times PLR_i + c_i \times PLR_i^2 + d_i \times PLR_i^3 \quad (3)$$

Equations (2) and (3) are chiller power consumption for Example 1 and Example 2 introduced from literature [8]. In (2) and (3), the coefficients  $a_i$ ,  $b_i$ ,  $c_i$ , and  $d_i$  define the relationship between power consumption and PLR for the chillers in Examples 1 and 2; this analysis was based on [8]. The PLR ranged between 0.0 and 1.0.

The overall objective of OCL optimization was to find the optimal partial loading rate of each chiller of the multi-chiller system that satisfied the cooling requirements of the venue but also minimized overall power consumption. Therefore, the objective function of the proposed OCL optimization was defined as shown in (4):

$$Obj_{Function} = \text{Minimize} \sum_{i=1}^n P_i \quad (4)$$

In (4), the parameter  $i$  signifies the  $i$ th chiller;  $n$  is the total number of chillers in the multi-chiller system;  $P_i$  is defined as the power consumption (kW) of the  $i$ th chiller. The object of Equation (4) is to find the lowest total power consumption in the multi-chiller system.

OCL problems have two types of restrictions [8] during solving; the first restriction is that the total output of RT must be equal to the RT required by the venue, as shown in (5).  $Q_i$  signifies the rated RT capacity of the  $i$ th chiller;  $CL$  is the total RT required by the venue. This is the basic constraint for

OCL problems. If the total RT generated by all chillers is smaller or larger than the RT requirement, the people who stay in the venue will feel not very comfortable.

$$\sum_{i=1}^n PLR_i \times Q_i = CL \quad (5)$$

The second restriction is that the partial loading of each chiller cannot be less than 30% [8], as defined in (6):

$$PLR_i \geq 0.3 \quad (6)$$

### 3. Two-Stage Differential Evolution Algorithm

The proposed two-stage DE algorithm is an optimization method that integrates MBDE and DE; it has superior exploration and convergence rates during searches for optimal solutions.

#### 3.1. Differential Evolution Algorithm

The DE (differential evolution) algorithm was proposed by Price and Storn in 1995 [18]. First, the initial real-number entities were randomly generated; the mutation operator used random numbers to select different entities to generate vector differences to serve as search directions, which were multiplied by a weight to obtain the size of the search, thus forming new search vectors. Then, through crossover, assessment, and selection, the process was iterated until the termination condition was met. The three main steps of the DE algorithm are: mutation, crossover, and selection.

##### 3.1.1. Mutation Operator

The purpose of the DE mutation operator is to generate different mutated vectors. Those vectors are generated by (7) and (8), where  $G$  represents the current iteration and  $G+1$  represents the next iteration.

$$V_{i,G+1} = X_{i,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (7)$$

$$V_{i,G+1} = X_{best,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (8)$$

From the group that ranges from Entity 1 to Entity  $P$ , randomly select two entities from whole DE group, the vectors  $X_{r2,G}$  and  $X_{r3,G}$ , perform subtraction between these two entities to find the difference vector, then multiply it by a given weight  $F$ . Finally, add the global best vector  $X_{best,G}$  or the self vector  $X_{i,G}$  to obtain the next generation of mutated vectors  $V_{i,G+1}$

##### 3.1.2. Crossover Operator

After an individual entity applies the mutation mechanism to generate a mutated vector, the crossover operator is used to generate trial vectors; the probability  $Cr$  of the crossover operator can range from 0.0 to 1.0.  $D$  represents the total design elements (or variables) in one vector. Regarding the randomly generated  $R$ , if the value of  $R$  is greater than  $Cr$ , then the  $j$ th element of original vector  $X_{ji,G}$  is selected; if the value of  $R$  is smaller than  $Cr$ , then the  $j$ th element of mutated vector  $V_{ji,G+1}$  is selected as the  $j$ th element of trial vector  $U_{ji,G+1}$  for the next generation. The crossover operator is formulated as shown in (9):

$$\text{if } R \leq Cr, U_{ji,G+1} = V_{ji,G+1} \quad \text{if } R > Cr, U_{ji,G+1} = X_{ji,G} \quad j = 1, 2, \dots, D \quad (9)$$

##### 3.1.3. Selection Operator

The selection operator is a convergence mechanism of the DE algorithm; it calculates the value of the objective function of the original vector  $X_{i,G}$  and the trial vector  $U_{i,G+1}$ , and compares the advantages and disadvantages of the objective function. The vector with the higher (or better) objective

function value becomes the next generation’s original vector  $X_{i,G+1}$ , then the vector with the highest objective function value of all original vector is selected as the global best vector  $X_{Best,G}$ .

### 3.2. Modified Binary Differential Evolution (MBDE) Algorithm

A modified DE algorithm [15,16] was proposed by Wu and Tseng in 2010. The real number encoding of the original DE algorithm was changed to a binary encoding, and a modified mutation operator was used for evolution. The method was also applied in engineering optimization problems [19]; the mechanism of the MBDE was the same as that of the original DE.

#### 3.2.1. Mutation Operator

The mutation equation for MBDE obtained by the original DE mutation operator [18] uses logical operators to compute binary strings as shown in (10) and (11).  $X_{i,G}$  is original vector and  $X_{r1,G}$  selects one entities randomly from the whole DE group.  $X_{best,G}$  represents the global best vector. This binary mutation uses the logical operator XOR (exclusive or) to divide the binary string into two groups, “0” and “1” strings. The “0” string group represents common characteristics between two entities; the set weight F2 and randomly generated value are compared; when the randomly generated value is smaller than F2, then the bit “0” or “1” is mutated to “1” or “0”. Alternatively, when the randomly generated value is greater than F2, then the bit remains unchanged. The “1” string group represents different characteristics between two entities; the given weight F1 and randomly generated value are compared; when the randomly generated value is smaller than F1, then the bit “0” or “1” is converted to “1” or “0”; alternately, when the randomly generated number is greater than F1, the bit remains unchanged. Finally, the two strings are combined to form the next generation of mutated vectors  $V_{i,G+1}$ ; the flowchart of the modified binary mutation operator is depicted in Figure 2.

$$V_{i,G+1} = F1(X_{i,G} \oplus X_{r1,G}) + F2!(X_{i,G} \oplus X_{r1,G}) \tag{10}$$

$$V_{i,G+1} = F1(X_{i,G} \oplus X_{best,G}) + F2!(X_{i,G} \oplus X_{best,G}) \tag{11}$$

Weight F1 tends to be greater than F2 because there is a higher probability that the characteristic of the optimal solution is a common characteristic, therefore a lower probability is used to maintain common characteristics, and a higher probability is used to change different characteristics.

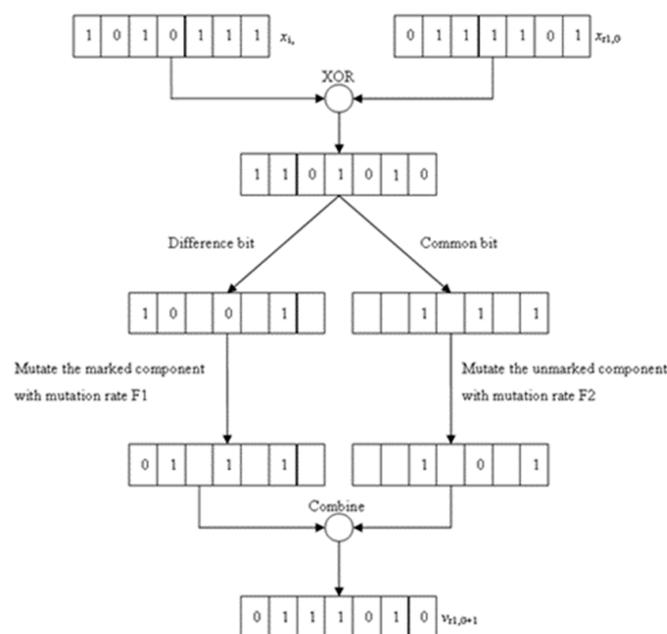


Figure 2. Modified binary mutation operator [15].

### 3.2.2. Crossover Operator

The binary crossover operator of the modified binary DE algorithm is similar to the crossover operator of the DE algorithm. For the mutually equal  $j$ th design variable, a randomly generated value and the set crossover rate are compared; if the randomly generated value is greater than the crossover rate, then the original vector is selected as the next-generation trial vector. Alternately, if the randomly generated value is smaller than the crossover rate, then the mutated vector becomes the next-generation trial vector, as shown in Figure 3.

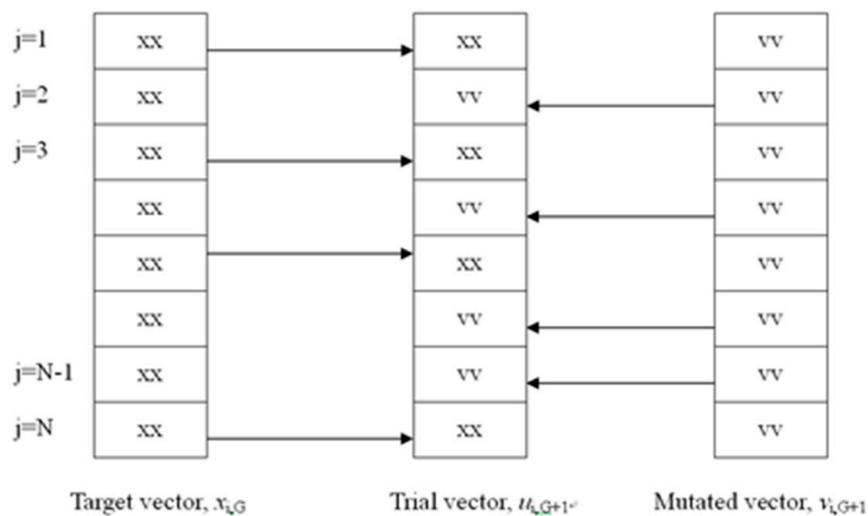


Figure 3. Binary crossover operator [15].

### 3.2.3. Selection Operator

The operation of selection is similar to the assessment and selection of DE. After binary mutation and binary crossover, the objective function  $F(X_{i,G})$  of the original vector is compared with the objective function  $F(U_{i,G+1})$  of the trial vector, and the best vector is selected as the next-generation objective vector  $X_{i,G+1}$ . The vector with highest objective function value among all DE group is selected as the global best vector  $X_{best,G}$  for mutation operator. The evolutionary computational system repeats Sections 3.2.1–3.2.3 iteratively until the termination condition is met.

### 3.3. Two-Stage Differential Evolution Algorithm

The two-stage DE algorithm solves OCL problems in two stages. In stage 1, the solving of MBDE begins after the PLR of each chiller has been encoded to binary, restrictions have been considered, and the objective equation has been defined. The character of binary encoding has low numerical precision but has favorable diversity when searching for an optimal solution [17]. After stage 1 has been solved, the resultant real-number encoding is taken into stage 2 of DE for exploitation. The overall architecture of the two-stage DE algorithm is illustrated in Figure 4.

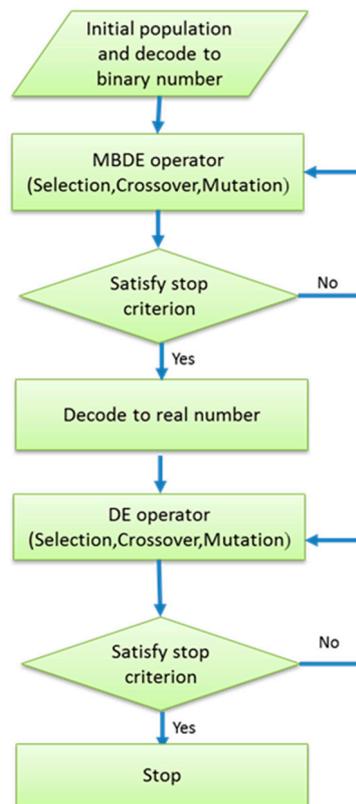


Figure 4. Flowchart of two-stage DE.

#### 4. Results, Analyses and Discussions

The OCL problems in [8] can serve as test examples for the proposed two-stage DE algorithm. Examples 1 and 2 are OCL problems for a six-chiller system and a four-chiller system. The parameters were set as follows: the population was 20; the mutation operator used (8) for DE and (11) for MBDE; the DE mutation rate and crossover rate were 0.5; the MBDE crossover rate was 0.5; the mutation rates F1 and F2 were 0.5 and 0.005, respectively. The aforementioned parameters were set in reference to the recommendations of [15,18]. The total number of iterations was configured to fit the required number of calculations of the example. The number of iterations in of case study 1 was 1000; it was 40 in case study 2.

##### 4.1. Case Study 1

This multi-chiller system was composed of six chillers. The objective was to calculate the lowest power consumption combinations of this six-chiller system under different RT requirements. The venue RT requirements in Example 1 were of five types: 6858 (90%); 6477 (85%); 6096 (80%); 5717 (75%); and 5334 (70%). Equation (2) defines the power consumption of this chiller system. The rated RT of each chiller and the power consumption parameters ( $a_i$ ,  $b_i$ ,  $c_i$ ) are listed in Table 1. The present study used a comparison method similar to those of other studies [8]; the total number of function evaluations was 20,000 for each run; 30 runs were executed, after which the mean, standard deviation (SD), maximum, and minimum values were computed in the manner of previously published articles in the literature.

Table 2 summarizes the optimization results of the proposed method with DCSA (differential cuckoo search approach) [8], showing that for both the minimum value and SD, all values of the optimal solution found by the two-stage DE algorithm under different CL conditions that were very close to those of DCSA. When CL was 80%, 75%, and 70%, the proposed method executed 20,000 function evaluations (2000 function evaluations for phase 1 and 18,000 function evaluations for phase 2) to

achieve better results than DCSA. Table 3 compared the optimal solutions of two-stage DE algorithm with those of other methods; when CL was 90% and 80%, a solution similar to that of DCSA was found; the solution was superior to than that of SA [20] and PSO [6]. Under other CL conditions (85%, 75%, and 70%), the results of the proposed two-stage DE algorithm were superior to those of other studies [6,8,20]. The convergence of two-DE algorithm for case study 1 is shown in Figure 5.

**Table 1.** Power consumption coefficient and rated RT information for case study 1.

Chiller	$a_i$	$b_i$	$c_i$	Rated RT
1	399.345	−122.12	206.30	1280
2	287.116	80.04	700.48	1280
3	−120.505	1525.99	−502.14	1280
4	−19.121	898.76	−98.15	1280
5	−95.029	1202.39	−352.16	1280
6	191.75	224.86	524.04	1280

**Table 2.** Case study 1: comparison with DCSA.

CL	Algorithm	MIN (kW)	Average (kW)	Max (kW)	SD
6868 (90%)	Two-stage DE	4738.575	4733.575	4738.575	$3.919 \times 10^{-6}$
	DCSA	4738.575	4738.575	4738.575	$5.313 \times 10^{-7}$
6477 (85%)	Two-stage DE	4421.649	4421.649	4421.650	$6.355 \times 10^{-5}$
	DCSA	4421.649	4421.650	4421.650	$2.301 \times 10^{-4}$
6096 (80%)	Two-stage DE	4143.706	4143.709	4143.714	$3.211 \times 10^{-4}$
	DCSA	4143.706	4143.710	4143.709	$4.299 \times 10^{-4}$
5717 (75%)	Two-stage DE	3838.208	3838.217	3838.225	$6.702 \times 10^{-4}$
	DCSA	3840.055	3840.458	3843.766	$9.428 \times 10^{-1}$
5334 (70%)	Two-stage DE	3507.270	3507.278	3507.302	$1.356 \times 10^{-3}$
	DCSA	3507.270	3507.715	3511.760	1.036

**Table 3.** Case study 1: comparison with methods proposed by other studies.

CL	Chiller No.	SA [20]	Power (kW)	PSO [6]	Power (kW)	DCSA [8]	Power (kW)	Two Stage DE	Power (kW)
		$PLR_i$	of	$PLR_i$	of	$PLR_i$	of	$PLR_i$	of
6898 (90%)	1	0.7789		0.8026		0.812726		0.81273	
	2	0.7587		0.7799		0.749619		0.749554	
	3	0.9791	4777.03	0.9996	4739.53	1.000000	4738.575	1.000000	4738.5750
	4	0.9781		0.9998		1.000000		1.000000	
	5	0.9820		0.9999		1.000000		1.000000	
	6	0.9265		0.8183		0.838559		0.838621	
6477 (85%)	1	0.8051		0.7606		0.727731		0.720409	
	2	0.6056		0.6555		0.656132		0.634290	
	3	0.9689	4453.67	1.0000	4423.04	1.000000	4421.649	1.000000	4421.6486
	4	0.9941		1.0000		1.000000		1.000000	
	5	0.9866		1.0000		1.000000		1.000000	
	6	0.7432		0.6835		0.716524		0.746387	
6096 (80%)	1	0.5635		0.6591		0.642735		0.642368	
	2	0.5743		0.5798		0.562645		0.562711	
	3	0.9675	4178.73	0.9991	4147.69	1.000000	4143.706	0.999999	4143.7064
	4	0.9798		0.9979		1.000000		0.999999	
	5	0.9845		0.9921		1.000000		0.999999	
	6	0.7338		0.5710		0.594490		0.594798	
5717 (75%)	1	0.6140		0.7713		0.843697		0.843243	
	2	0.4429		0.7177		0.783794		0.783222	
	3	0.9891	3925.51	0.3000	3921.07	0.000001	3840.055	0.000000	3838.2079
	4	0.8867		0.9991		1.000000		0.999999	
	5	0.9841		1.0000		1.000000		0.999999	
	6	0.5875		0.7187		0.883049		0.882499	
5334 (70%)	1	0.6265		0.6418		0.749969		0.758176	
	2	0.7403		0.6621		0.682477		0.689668	
	3	0.3093	3675.34	0.3301	3642.55	0.000012	3507.270	0.000000	3507.269
	4	0.9546		0.9906		1.000000		1.000000	
	5	0.9511		0.9990		1.000000		1.000000	
	6	0.6250		0.5806		0.776363		0.760606	

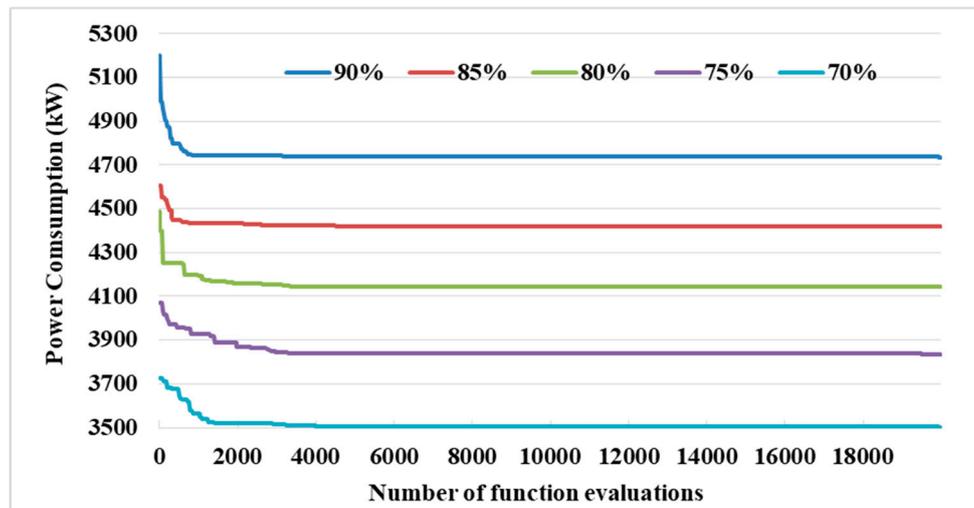


Figure 5. Convergence of two-stage DE for case study 1.

#### 4.2. Case Study 2

This multi-chiller system was composed of four chillers. The objective of this example was the same as that of Example 1, namely, to find the lowest power consumption combination for this four-chiller system at different RT requirements. The RT requirements of the venue were of six types: 2610 (90%); 2320 (80%); 2030 (70%); 1740 (60%); 1450 (50%); and 1160 (40%). Equation (3) defines the power consumption of the chiller, the rated RT of each chiller and their power consumption parameters ( $a_i$ ,  $b_i$ ,  $c_i$ ,  $d_i$ ) are listed in Table 4. The study referenced the comparison methods of relevant articles [4,6–8], and 800 function evaluations were allowed in each run, and 30 runs were executed. The mean, SD, maximum, and minimum values were calculated for comparison with those of other studies.

Table 4. Case study 2: Power consumption coefficient and rated RT information.

Chiller	$a_i$	$b_i$	$c_i$	$d_i$	Rated RT
1	104.09	166.57	−430.13	512.53	450
2	−67.15	1177.79	−2174.53	1456.53	450
3	384.71	−779.13	1151.42	−63.2	1000
4	541.63	413.48	−3626.5	4021.41	1000

Table 5 proves that in addition to the power consumption of the proposed two-stage DE algorithm being relatively similar to that of DCSA at CL = 80%, under other CL conditions (70%, 60%, 50%, and 40%), the results of the proposed optimization method using 800 function evaluations (80 function evaluations for phase 1 and 720 function evaluations for phase 2) were also superior to that of DCSA [8]. In particular, at CL = 50% and CL = 40%, the SD results validated that the proposed two-stage DE algorithm was more stable than DCSA. The results of comparisons with other methods [4,6–8] are listed in Table 6, they prove that the proposed two-stage DE algorithm could find a power consumption combination that was similar to or lower than those of other methods (GA [4], PSO [6], DE [7], CSA [8]) under most CL conditions. The convergence of two-DE algorithm for case study 2 is shown in Figure 6.

Table 5. Case study 2: comparison with DCSA.

CL	Algorithm	MIN (kW)	Average (kW)	Max (kW)	SD
2610 (90%)	Two-Stage DE	1857.297	1858.031	1859.626	$1.317 \times 10^{-1}$
	DCSA	1857.299	1857.315	1857.401	$2.329 \times 10^{-2}$
2320 (80%)	Two-Stage DE	1458.334	1458.344	1458.931	$2.130 \times 10^{-2}$
	DCSA	1455.665	1455.810	1458.478	$5.303 \times 10^{-1}$
2030 (70%)	Two-Stage DE	1178.138	1178.928	1182.952	0.2006
	DCSA	1178.137	1181.067	1199.495	4.803
1740 (60%)	Two-Stage DE	942.050	976.182	1001.170	4.9418
	DCSA	942.183	972.076	1008.493	25.721
1450 (50%)	Two-Stage DE	752.963	759.612	792.907	1.851
	DCSA	753.004	765.340	824.347	17.429
1160 (40%)	Two-Stage DE	583.938	630.990	661.460	3.985
	DCSA	583.923	644.933	726.016	44.015

Table 6. Case study 2: comparison with the methods proposed by other studies.

CL Chiller No.	GA [4]	Power (kW)	DE [7]	Power (kW)	PSO [6]	Power (kW)	DCSA [8]	Power (kW)	Two Stage DE	Power (kW)
i	PLR <sub>i</sub>	of								
2610 (90%)	1	0.990000		0.990000		0.990000		0.990988		0.990491
	2	0.950000	1862.18	0.910000	1857.30	0.910000	1857.30	0.905473	1857.299	0.905503
	3	1.000000		1.000000		1.000000		1.000000		1.000000
	4	0.740000		0.760000		0.760000		0.756593		0.756791
2320 (85%)	1	0.860000		0.830000		0.830000		0.828756		0.822981
	2	0.810000	1457.23	0.810000	1455.66	0.810000	1455.66	0.805457	1455.665	0.801856
	3	0.880000		0.900000		0.900000		0.896722		0.885369
	4	0.690000		0.690000		0.690000		0.687883		0.685549
2030 (70%)	1	0.660000		0.730000		0.730000		0.773478		0.725289
	2	0.760000	1183.80	0.740000	1178.14	0.740000	1178.14	0.739801	1178.137	0.739752
	3	0.760000		0.720000		0.720000		0.721146		0.722185
	4	0.640000		0.650000		0.650000		0.627878		0.648549
1740 (60%)	1	0.600000		0.600000		0.600000		0.767678		0.745135
	2	0.700000	1001.62	0.660000	998.53	0.660000	998.53	0.004531	942.183	0.000000
	3	0.510000		0.560000		0.560000		0.746317		0.748647
	4	0.590000		0.610000		0.610000		0.646189		0.656017
1450 (50%)	1	0.600000		0.610000		0.610000		0.515832		0.599201
	2	0.360000	907.72	0.000000	820.07	0.000000	820.07	0.000001	753.004	0.000000
	4	0.440000		0.570000		0.570000		0.610547		0.571431
	4	0.580000		0.610000		0.610000		0.607328		0.656017
1160 (40%)	1	0.330000		0.000000		0.000000		0.000000		0.000000
	2	0.320000	856.30	0.000000	651.07	0.000000	651.07	0.000014	583.923	0.000012
	3	0.320000		0.560000		0.560000		0.570369		0.556082
	4	0.540000		0.600000		0.600000		0.589625		0.603912

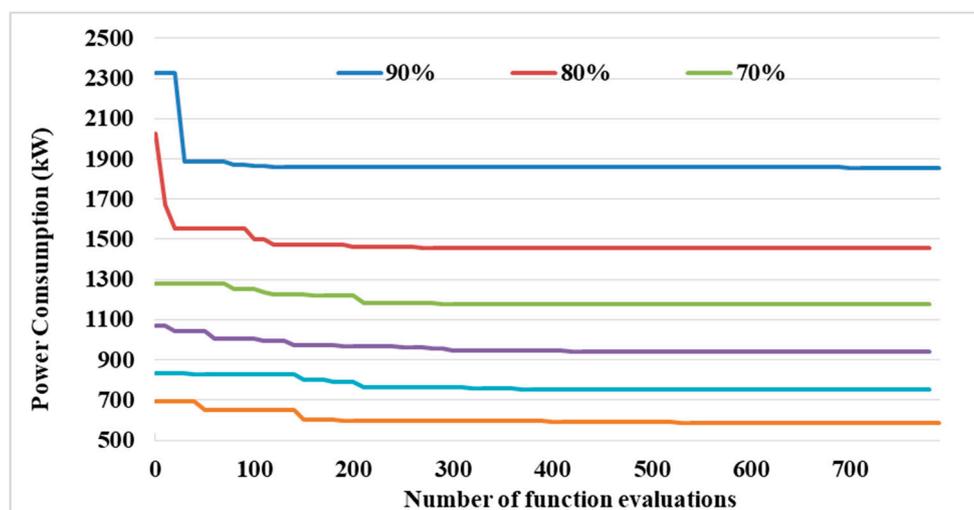


Figure 6. Convergence of two-stage DE for case study 1.

## 5. Conclusions

The proposed two-stage DE algorithm integrates the characteristics of binary and real-valued DE algorithms; it has excellent exploration and convergence rates for optimal solutions. Regarding case studies 1 and 2, the comparisons of the proposed method with other methods indicate that the proposed two-stage DE algorithm is suitable for optimization of the power consumption configurations of multi-chiller systems. In addition to being able to obtain solutions similar to or better than those of referenced studies, the degree of variance of the optimal solution in each search was better than those of other studies, further validating that the performance and stability of the proposed two-stage DE algorithm are better than those of other methods. The proposed algorithm can be used to solve similar optimization problems.

**Author Contributions:** Conceptualization: C.-M.L., C.-C.K. and K.-Y.T.; methodology: K.-Y.T.; software: C.-C.K.; validation: C.-M.L., C.-Y.W. and S.-F.L.; investigation: C.-M.L.; data curation: K.-Y.T.; writing—original draft preparation: C.-M.L.; writing—review and editing: C.-Y.W. and S.-F.L.

**Funding:** This research was funded by the Bureau of Energy, Ministry of Economic Affairs, R.O.C. (Taiwan).

**Acknowledgments:** The authors would like to acknowledge the support from Energy Fund of Ministry of Economics Affairs, Taiwan, National Chiao Tung University and Tatung University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Schwedler, M.; Yates, A. *Applications Engineering Manual. Multiple-Chiller-System Design and Control*; Trane: Dublin, Ireland, 2001.
- Chang, Y.C. A novel energy conservation method—Optimal chiller loading. *Electr. Power Syst. Res.* **2004**, *69*, 221–226. [[CrossRef](#)]
- Chang, Y.C.; Lin, F.A.; Lin, C.H. Optimal chiller sequencing by branch and bound method for saving energy. *Energy Convers. Manag.* **2005**, *46*, 2158–2172. [[CrossRef](#)]
- Chang, Y.C. Genetic algorithm based optimal chiller loading for energy conservation. *Appl. Therm. Eng.* **2005**, *25*, 2800–2815. [[CrossRef](#)]
- Chang, Y.C. Optimal chiller loading by evolution strategy for saving energy. *Energy Build.* **2007**, *39*, 437–444. [[CrossRef](#)]
- Ardakani, A.J.; Ardakani, F.F.; Hosseinian, S.H. A novel approach for optimal chiller loading using particle swarm optimization. *Energy Build.* **2008**, *40*, 2177–2187. [[CrossRef](#)]
- Lee, W.S.; Chen, Y.T.; Kao, T. Optimal chiller loading by differential evolution algorithm for reducing energy consumption. *Energy Build.* **2011**, *43*, 599–604. [[CrossRef](#)]
- Coelho, L.D.S.; Klein, C.E.; Sabat, S.L.; Mariani, V.C. Optimal chiller loading for energy conservation using a new differential cuckoo search approach. *Energy* **2014**, *75*, 237–243. [[CrossRef](#)]
- Tan, K.C.; Chiam, S.C.; Mamun, A.A.; Goh, C.K. Balancing exploration and exploitation with adaptive variation for evolutionary multiobjective optimization. *Eur. J. Oper. Res.* **2009**, *197*, 701–713. [[CrossRef](#)]
- Binkley, K.J.; Hagiwara, M. Balancing exploration and exploitation in Particle Swarm Optimization: Velocity-based Reinitialization. *Inf. Media Technol.* **2008**, *3*, 103–111. [[CrossRef](#)]
- Epitropakis, M.G.; Plagianakos, V.P.; Vrahatis, M.N. Balancing the exploration and exploitation capabilities of the Differential Evolution Algorithm. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 2686–2693.
- Bao, H. A Two-Phase Hybrid Optimization Algorithm for Solving Complex Optimization Problems. *Int. J. Smart Home* **2015**, *9*, 27–36. [[CrossRef](#)]
- Beghi, A.; Cecchinato, L.; Rampazzo, M. A multiphase genetic algorithm for the efficient management of multichiller systems. *Energy Convers. Manag.* **2011**, *52*, 1650–1661. [[CrossRef](#)]
- Cheng, M.; Tran, D. Two-Phase Differential Evolution for the Multiobjective Optimization of Time–Cost Tradeoffs in Resource-Constrained Construction Projects. *IEEE Trans. Eng. Manag.* **2014**, *61*, 450–461. [[CrossRef](#)]

15. Wu, C.Y.; Tseng, K.Y. Topology optimization of structures using modified binary differential evolution. *Struct. Multidiscip. Optim.* **2010**, *42*, 939–953. [[CrossRef](#)]
16. Wu, C.Y.; Tseng, K.Y. Stress-based binary differential evolution for topology optimization of structures. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2010**, *224*, 443–457. [[CrossRef](#)]
17. Okabe, T.; Jin, Y.; Sendhoff, B. Combination of Genetic Algorithms and Evolution Strategies with Self-adaptive Switching. In *Multi-Objective Memetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 281–307.
18. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
19. Wu, C.Y.; Tseng, K.Y. Engineering Optimization Using Modified Binary Differential Evolution Algorithm. In Proceedings of the Third International Joint Conference on Computational Science and Optimization, Huangshan, China, 28–31 May 2010; pp. 501–505.
20. Chang, Y.C. An innovative approach for demand side management—Optimal chiller loading by simulated annealing. *Energy* **2006**, *31*, 1883–1896. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).