



Optimal chiller loading in HVAC System Using a Novel Algorithm Based on the distributed framework

Junqi Yu, Qite Liu, Anjun Zhao^{*}, Xuegen Qian, Rui Zhang

Xi'an University of Architecture and Technology, Shaanxi, Xi'an, China

ARTICLE INFO

Keywords:

Optimal chiller loading
Distributed framework
Estimation of distribution algorithm
Energy saving

ABSTRACT

Aimed at the optimal chiller loading (OCL) problem in heating, ventilation and air conditioning (HVAC) system, a distributed framework for HVAC control is introduced and discussed. And furthermore, the distributed chaotic estimation of distribution algorithm (DCEDA) based on this framework is proposed. Firstly, compared with the centralized framework, the distributed framework has the features of flexibility and expansibility. Therefore, the distributed framework can fit the development of HVAC control system better. Secondly, in the proposed algorithm, an initialization methodology based on logistic map is developed and the chaotic mutation mechanism is applied to increase the search capability of the algorithm. To testify the performance of DCEDA, two well-known cases based on the OCL problem are tested and the results are compared with other algorithms. The results show that DCEDA is an efficient distributed optimization algorithm with good robustness, stability and convergence, and it can achieve significant energy saving effect.

1. Introduction

1.1. Background and motivation

Recently, multi-chiller system is widely used in large public buildings. As major components of heating, ventilation and air conditioning (HVAC) system, chillers consumed about 25–40% of the total electricity consumption in a commercial building [1]. However, chillers have different performance features and capacities, which can increase operating flexibility, and minimize energy consumption under the condition of meeting demands of different loads [2]. Therefore, operating the multi-chiller system to minimize the energy consumption under different load requirements, which known as the problem of optimal chiller loading (OCL), has attracted extensive attention from researchers nowadays [3].

Nevertheless, the HVAC system is highly complex and nonlinear, and the traditional centralized control system of it also has the characteristics of large-scale spatial distribution and multi-device cooperation. Therefore, there are some deficiencies in its construction and operation. Firstly, network construction is a complex and time-consuming process, because lots of secondary development, such as configuration and debugging, must be completed by professional engineers on the construction site. Secondly, in some key procedure, such as multi-chiller

control and multi-pump control, the algorithms and models must be written into the centralized controller. However, these algorithms and models need to be reprogrammed case by case owing to the changes of system configuration or device type. Thirdly, the accurate and detailed performance parameters of device, which are needed to establish the system model and the optimization algorithm, are difficult to acquire. Thus, the system integrators usually build algorithms and models relying on the easily available inexact parameters, which leads to a less than optimum control effect. More importantly, the centralized method needs to transfer the information to the central controller for computation and control. Once the central controller fails, the whole system cannot work properly. What's more, with the continuous expansion of HVAC system, the centralized control system will make greater demands on the central controller. Owing to the above defects, although many centralized optimization algorithms have been developed for HVAC system, only a few of them have been applied in practical projects. In some senses, the conventional centralized framework has become an obstacle to the development of the HVAC control system.

Over the last decade, the distributed framework has gained growing renewed interest owing to its various applications in power systems, communication networks, machine learning, and sensor networks [4]. In contrast with the centralized framework, the algorithm based on the distributed framework achieves optimal decisions by the local manipulation with private data and the diffusion of local information through

^{*} Corresponding author.

E-mail address: zhao_anjun@163.com (A. Zhao).

<https://doi.org/10.1016/j.job.2019.101044>

Received 10 June 2019; Received in revised form 30 October 2019; Accepted 31 October 2019

Available online 2 November 2019

2352-7102/© 2019 Elsevier Ltd. All rights reserved.

Nomenclature

a, b, c, d	performance coefficients of chiller
C	mutation step size
Cmin	minimum variation step size
fit(i)	fitness value
fit'(i)	linear fitness value of the ith individual
flag	iterative termination signal
Gmax	maximum number of iterations
high	maximum value of the variable
K	new population multiplication
low	minimum value of the variable
M	number of better individuals
N	number of population size
Pchiller	the power consumed by the chiller

penal	penalty factor
PLR	part load ratio
pop(i)	the ith individual
pop'(i)	the ith individual after chaotic mutation operation
position(i)	the position of the ith individual in the order of fitness from small to large
Qi0	the rated cooling capacity of the ith chiller
Qneed	terminal load requirement
r(i)	the chaotic variation radius of the ith individual
Wtotal	the total power consumed by the parallel chiller system
α_i	the value of the ith bit of a chaotic sequence
β	parameter of one-dimensional logistic mapping
μ	the mean of the gaussian distribution model
σ	the variance of gaussian distribution model

a multi-agent network [5]. In practice, the distributed optimization can be integrated naturally with large-scale networks and complex systems owing to its scalability and distributive nature. Compared with centralized scheme, the distributed optimization strategies, which offer robustness and flexibility in general, can fit seamlessly in multi-agent system in particular [4]. While, the HVAC system is also a multi-agent network, so the distributed framework and distributed optimization algorithm should be more suitable for the control of HVAC system.

In addition, estimation of distribution algorithm (EDA) is a stochastic optimization algorithm based on statistical principle. EDA establishes a probability model describing the spatial distribution of the optimal solution from a macroscopic point of view by using statistical learning method, which is different from genetic algorithm (GA). Then the probability model is randomly sampled to generate a new population, so that the population can evolve gradually to the optimal solution [6]. Recently, EDA and its variants have been used to solve various optimization problems [7–11], and they all obtained the competitive results compared to other methods. So, EDA has the potential to solve the OCL problem effectively.

Above all, the distributed optimization algorithm which is improved by the EDA algorithm, could have some potential of solving OCL problem.

1.2. Objectives

The main objective of this study is to propose a distributed optimization algorithm based on the distributed framework and apply it to solve the OCL problem in the HVAC system. The proposed algorithm is called distributed chaotic estimation of distribution algorithm (DCEDA), which is improved by the EDA algorithm and the chaotic mutation operator. This study is intended as a case that will contribute to promoting the implementation of distributed framework and distributed algorithm in the HVAC system.

The objectives of this paper are as follows:

- (1) Introduce and discuss the distributed framework of HVAC control system.
- (2) Propose a distributed optimization algorithm called DCEDA based on the distributed framework.
- (3) Apply the proposed algorithm to solve the OCL problem in the HVAC system.
- (4) Search the best parameters of the proposed algorithm in solving the OCL problem.

This section has briefly introduced the background, motivation and objectives of the paper. The remainder of the paper is organized as follows. In Section 2, the previous work about the algorithm for solving

the OCL problem and the application of the distributed framework are discussed. In Section 3, the OCL problem is described. In Section 4, the distributed framework and the implementation of DCEDA on OCL problem are described in detail. In Section 5, a description of two cases study is presented. In Section 6, the experiment results are analyzed. And in Section 7, conclusions are drawn.

2. Literature review

2.1. Algorithm for solving the OCL problem

Recently, lots of optimization algorithms have been applied to solve the OCL problem. They can be classified into two categories, the exact algorithm and the meta-heuristic algorithm. Chang [12] used Lagrange method (LM) to solve the OCL problem, but the experimental results showed that the algorithm may not converge at the lower load demand. Therefore, Chang proposed the gradient method (GM) in Ref. [13] to solve the problem of economic dispatch of chillers. This method could overcome the problem that the LM cannot converge at the lower load demand, but its solution accuracy was slightly worse than LM. Later, the generalized reduced gradient (GRG) method was also adopted to minimize the energy consumption of the system at different cooling loads, and it was compared with other algorithms in Ref. [14].

In addition, many meta-heuristic algorithms were also proposed to solve realistic optimization problems, such as improved firefly algorithm (IFA) [15], differential search (DS) algorithm [16], differential cuckoo search algorithm (DCSA) [17], bacterial foraging optimization algorithm (BFOA) [18], chaos mutation firefly algorithm (CMFA) [19], coyote optimization algorithm (COA) [20], farmland fertility algorithm [21], pity beetle algorithm (PBA) [22], exchange market algorithm (EMA) [23], improved invasive weed optimization (EIWO) algorithm [2], improved artificial fish swarm algorithm (VAFSA) [24], enhanced artificial bee colony algorithm (EABC) [25], artificial neural network combined with artificial bee colony algorithm (FF-ABC) [26], enhanced crow search algorithm (ECSA) [27], whale optimization algorithm (WOA) [28], multi-agent shuffled frog leaping algorithm (MASFLA) [29], improved chaotic invasive weed optimization algorithm [30], improved fruit fly optimization algorithm (IFOA) [31], and improved social spider optimization algorithm (NISSO) [32]. Among these meta-heuristic algorithms, several have been used to solve the OCL problem. Chang [33,34] solved the OCL problem by GA with binary coding, and it also solved LM's problem of not being able to deal with a system with non-convex function. Lee [35] and Ardakani [36] applied particle swarm optimization (PSO) and continuous GA (CGA) to solve the OCL problem. Chen [37] used neural networks (NN) to build the model of power consumption of the chiller and adopted PSO algorithm to optimize the chiller loading to reduce the power consumption of the

system. The results showed that PSO and CGA outperformed than GA in finding optimal solution and they could also overcome the divergence problem caused by the LM occurring at low demand. Lee [3] adopted the differential evolution (DE) algorithm to solve the OCL problem, and DE could find the equal optimal solution compared with PSO. Coelho [15, 17] proposed the IFA and DCSA to solve the OCL problem and they all reduced the energy consumption of the system compared with other early algorithms. The teaching-learning-based optimization (TLBO) algorithm proposed in Refs. [38,39] and EIWO proposed in Ref. [2] were also the effective and efficient method for solving the OCL problem. Later, Zheng [24] applied the VAFSA to minimize the total power energy consumption of chillers and cooling towers in the HVAC system. The experimental results showed that VAFSA could save power energy with the well convergent ability. Hamid Teimourzadeh [40] implemented the augmented group search optimization (AGSO) algorithm on the economic cooling load dispatch of multi-chiller plants. Numerical results demonstrated that AGSO could achieve a lower energy consumption than that of recently published algorithms with higher convergence speed. It can be concluded from the presented algorithms that the meta-heuristic algorithms have shown the efficiency for solving the OCL problem.

2.2. The application of the distributed framework

In recent years, the optimization algorithm based on the distributed framework has been widely investigated via multi-agent networks. In the process of distributed optimization, the global cost function is formed by multiple local cost functions [41]. To find the global minimum value of the global cost function, each agent minimizes its own cost function by exchanging local information with other agents via a communication network. And with distributed computing enabled by distributed algorithms, the heavy computational burden will be split among and dispersed over different computational units in order to achieve computational efficiency. Nowadays, the distributed optimization algorithm is normally used in economic dispatching of power systems [42,43], automatic coordination of vehicles [44,45], and dynamic task allocation in a robotic swarm [46] and so on, and all of them have achieved good results. All the above application scenarios have the characteristics of large distributed network and multi-agent system structure. While, in the field of HVAC system, which also has the same characteristics, the application of distributed optimization algorithm is rare.

3. Problem description

In the control of HVAC system, the OCL problem is to reduce the total energy consumption of the system as far as possible to achieve the purpose of energy saving under the condition of meeting the terminal cooling load demand. Since the power of chiller is related to its part load ratio (PLR), the power consumed by the chiller is usually synthesized into the polynomial form of PLR as shown in Eq. (1) [2,3,14–17,23,24, 33–40], where a , b , c and d represent the performance coefficients of the chiller.

$$P_{chiller} = a + b \times PLR + c \times PLR^2 + d \times PLR^3 \quad (1)$$

For the OCL problem, the minimum energy consumption of the multi-chiller system is considered as the optimization objective, the terminal cooling load demand is selected as the constraint, the PLR of each chiller is as the optimization variable, and the optimization model is set up. So, its optimization objective can be described by the mathematical formula as shown in Eq. (2), where W_{total} is the total power consumed by the parallel chiller system, Q_i^0 is the rated cooling capacity of the i th chiller, and Q_{need} is the terminal load requirement.

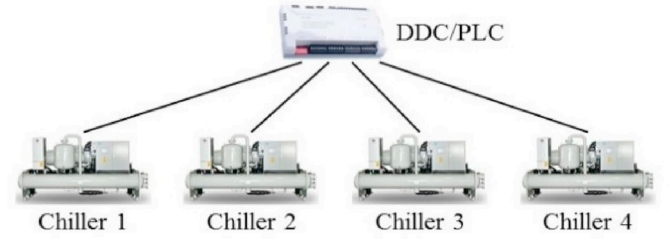


Fig. 1. Centralized framework of multi-chiller system.

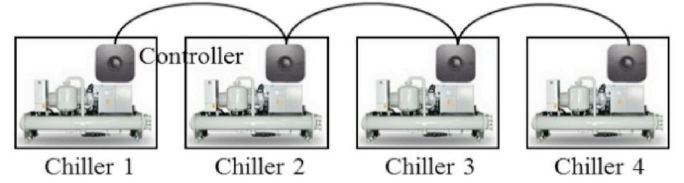


Fig. 2. Distributed framework of multi-chiller system.

$$\begin{aligned} \min (W_{total}) \\ s.t. \sum_{i=1}^N PLR_i \times Q_i^0 = Q_{need} \quad , \quad i = 1, 2, \dots, N \end{aligned} \quad (2)$$

Considering the performance of the chiller and the suggestions of the manufacturer, the PLR of each chiller should not be less than 0.3. Then, the objective function and constraint conditions for the optimization of multi-chiller system can be listed as shown in Eq. (3).

$$\begin{aligned} \min (W_{total}) \\ s.t. \quad 0.3 \leq PLR_i \leq 1 \quad \text{or} \quad PLR_i = 0 \\ \sum_{i=1}^N PLR_i \times Q_i^0 = Q_{need} \quad , \quad i = 1, 2, \dots, N \end{aligned} \quad (3)$$

4. Methodology

4.1. The distributed framework

As is shown in Fig. 1, in the centralized framework, each chiller is connected to the centralized controller, which performs calculations and makes decisions based on the centralized algorithm and sends out control instructions to each chiller. This framework inevitably has some drawbacks. For example, inaccurate equipment performance parameters lead to a less than optimum control effect, complex and time-consuming field configurations increase labor costs, and control algorithms need to be developed case by case. In addition, these are also the reasons why the developed centralized algorithms are difficult to be widely applied in practical projects.

However, in the distributed framework shown in Fig. 2, device such as chiller is upgraded to an agent by installing a distributed controller that contains detailed and accurate performance parameters written in a uniform format by the manufacturer who knows the performance of the device best. In the field installation process, all agents only need to be connected to form a control network via wired or wireless communication according to the actual physical topology. In the way above, the control network becomes a multi-agent network without central controller. In addition, each agent in the network has an equal status, and any agent can be regarded as a central controller.

After the distributed control network is completed, the required distributed optimization algorithm can be downloaded to any agent, and then spread to all agents by means of neighbor communication. In this way, the algorithms are identical within the same type of device and are portable rather than developed case-by-case. When optimization is carried out, agents can communicate and negotiate with neighbors

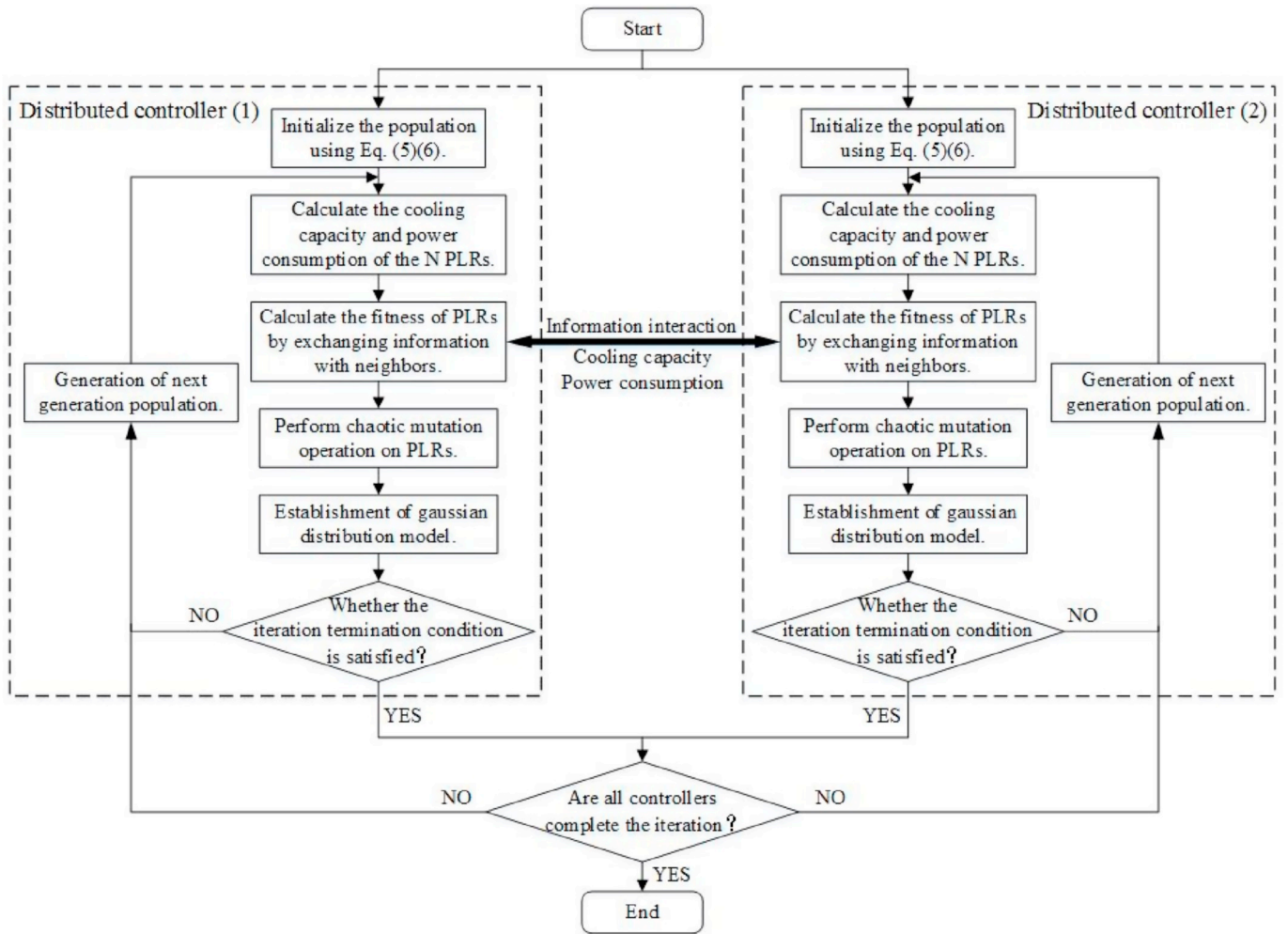


Fig. 3. The flow chart of the DCEDA algorithm with two distributed controllers as examples.

according to the design of the algorithm, send local variables to its neighbors and accept related variables from them, and work together to meet the control requirements of the system to achieve the goal of energy saving. Therefore, under this framework, since there is no need for complex system modeling and algorithm development, the field configuration of control network will be greatly simplified, and the system will be more flexible and extensible.

4.2. Implementation of DCEDA on OCL problem

EDA can solve the low-dimensional problem well, but the solution of the high-dimensional problem, such as the OCL problem, often falls into the local optimal values, which leads to premature convergence. The reason is that EDA does not introduce the mutation operator. With the continuous iteration of the algorithm, the diversity of its population will gradually decrease, which will result in premature convergence easily. In view of this shortcoming, the chaotic mutation operator is applied into the traditional EDA to improve the overall performance of the algorithm by increasing the diversity of the population.

Chaotic state exists widely in nature and human society and is a common phenomenon in nonlinear system. Chaotic motion has the characteristics of randomness, ergodicity and regularity, which determines that it can traverse all states within a certain range without repeating. And the chaotic mutation operator makes use of these features to ensure the diversity of the population. The main idea of the proposed algorithm is to introduce chaotic mutation operator into the traditional EDA and then improve it into a distributed optimization

algorithm.

When solving the OCL problem, the optimization objective is the minimum energy consumption of the multi-chiller system, the optimization parameter is the PLR of each chiller, and the constraint is the terminal cooling load demand. If the centralized framework is adopted, a central controller is required to collect the global information for optimization computation to obtain the control strategy, and send the control signals to each chiller finally. However, under the distributed control framework, all distributed controllers have the same distributed optimization algorithm built in. Each agent carries out optimization calculation based on the information transmitted by its neighbors, makes decisions and controls the running status of the device it connected. Finally, the system achieves the goal of energy conservation. Taking two distributed controllers as an example, the flow chart of the DCEDA when solving the OCL problem is shown in Fig. 3, and the key steps are shown below.

4.2.1. Encoding and initialization

In the proposed algorithm, the initial population of each distributed controller is generated based on one-dimensional logistic map shown in Eq. (4) [47], and extended to the variation range of variables through Eq. (5), where *low* and *high* represent the minimum and maximum values of the variable $pop(i)$ respectively. In order to find the optimal solution of the problem which needs to be solved, more search space should be covered by the initial population as much as possible. When $\beta = 4$, the values of α_i would take any real number between 0 and 1 and never repeat a value having turned up already [48], which can expand the

search range. So, take $\beta = 4$, and α_i is a random number within 0–1.

In solving the OCL problem, the variable is the PLR of each chiller, the range of PLR is shown in Eq. (3), and $low = 0$, $high = 1$, when $pop(i) < 0.3$, $pop(i) = 0$.

$$\alpha_{i+1} = \beta \alpha_i (1 - \alpha_i) \quad (4)$$

$$pop(i) = low + (high - low) \times \alpha_i \quad (5)$$

4.2.2. Fitness function

Under the distributed framework, the fitness value of the individual in each controller is calculated according to its own information and the interaction variables transmitted from its neighbors. Since the OCL problem is an optimization problem with constraints, the penalty function is adopted. Penalty function is one of the most popular methods used to solve the optimization problem with constraint conditions [2, 15–17, 24, 33, 36]. With the penalty function, the constrained objective function can be transformed into unconstrained objective function. If the individual obtains a better objective function value while satisfying the restricting conditions, the fitness value of it will be higher. If the individual can't satisfy the restricting conditions, which represents an infeasible solution, a large enough penalty will be given to reduce its fitness. So, the penalty strategy can decrease the fitness value of the infeasible solution.

When solving the OCL problem, the fitness function is constructed as shown in Eq. (6), where $penal$ is the penalty factor. The interaction variables are cooling capacity and energy consumption of the chiller. And the restricting condition is that the sum of the cooling capacity of all chillers is equal to the demand of terminal cooling load. If the operation strategy of the j th individual can't fulfill the system cooling load requirement, a large enough $penal$ will be given. After that, the $fit(j)$ becomes smaller than other individuals which can satisfy the requirement. If the cooling capacity can fulfill the load demand, the lower the system energy consumption of the j th individual is, the higher the $fit(j)$ is.

$$fit = 1 / \left(W_{total} + penal \times \left(\sum_{i=1}^N PLR_i \times Q_i^0 - Q_{need} \right)^2 \right) \quad (6)$$

4.2.3. Chaotic mutation

The mutation operator is one of the effective methods to increase and maintain the diversity of the population, and is meanwhile an efficient method to escape the local optimum solution and to overcome the premature convergence. In this paper, the chaotic mutation operator is adopted. Firstly, individuals are ranked according to the fitness from smallest to largest. Then the linear fitness of individual is calculated according to the linear allocation method. The linear fitness value of the i th individual in the population is shown in Eq. (7) [49], where N is the population size, $position(i)$ is the position where the i th individual is ranked from smallest to largest according to the fitness value

$$fit^*(i) = position(i) / N \quad (7)$$

Secondly, the variation radius of the chaotic mutation operator is determined. In order to make the algorithm converge faster and better, the fitness values of individuals are considered during the chaotic mutation operation. That is, the variation radius of individual with small fitness is increased, while that of individual with large fitness is reduced. Therefore, the variation radius of the i th individual in the population is determined by Eq. (8) [49]. Where C is the variation step size, and its value will affect the search performance of the algorithm. In the initial stage of the algorithm execution, a larger C is needed to expand the search scope, while in the later stage of evolution, a smaller C is required to realize the refined search of decision variables within a small range of optimal solution. Therefore, the value of C is determined according to the real time information of the population in the evolutionary process. The C in each iteration is calculated by Eq. (9) [49]. In order to ensure

the randomness of the algorithm, when $C < C_{min}$, let $C = C_{min}$ to increase the probability of the algorithm jumping out of the range of local optimal solutions.

$$r(i) = C \times (1 - fit^*(i)) \quad (8)$$

$$C = \frac{\max_{j=1,2,\dots,N} (pop(j)) - \min_{k=1,2,\dots,N} (pop(k))}{1.5} \quad (9)$$

Then, the chaotic mutation operator is performed on all individuals. The equation of chaotic mutation is mathematically described as Eq. (10) [49], where $pop(i)$ is the i th individual before mutation and $pop'(i)$ is the i th individual after mutation. α_i is a chaotic variable generated by Eq. (4), and $(-2 + 4\alpha_i)$ is the implementation of Eq. (5) to extend the variation range of chaotic sequence to $[-2, 2]$, so that the variation range of chaotic mutation is consistent with the traditional gaussian mutation [49]. Finally, fitness values of individuals before and after mutation are compared. If $fit(pop'(i)) > fit(pop(i))$, $pop'(i)$ is used to replace the original $pop(i)$ to obtain a new population. Therefore, the chaotic mutation operator can be described as taking a random value as a new individual within the radius of $2r$ of the original individual. If the fitness of the new individual is greater than that of the original individual, the original individual in the population will be replaced with the new individual. This mutation operation not only ensures that the fitness of the mutated population is greater than the initial population, but also increases the probability of the population escaping from the local optimal solution. In other words, the population can continuously evolve to the optimal solution through this mutation operation.

$$pop'(i) = pop(i) + r(i) \times (-2 + 4\alpha_i) \quad (10)$$

4.2.4. Establishment of Gaussian distribution model

After the chaotic mutation operation is completed, the fitness of the population in each controller is calculated, and M superior individuals are selected as samples for statistical analysis, and the mean and variance of the gaussian distribution are obtained by Eq. (11).

$$\begin{cases} \mu = \frac{\sum_{j=1}^M pop(j)}{M} \\ \sigma = \sqrt{\frac{\sum_{j=1}^M (pop(j) - \mu)^2}{M}} \end{cases} \quad (11)$$

4.2.5. Generation of next generation population

The distributed controller generates $K \times N$ individuals subject to gaussian distribution model according to the calculated sample mean and variance, and combines them with M superior parent individuals as a whole. Then the fitness of $K \times N + M$ individuals are evaluated, and N superior individuals are selected as the next generation population.

4.2.6. Iteration termination condition

Each distributed controller needs to make judgments at the end of each evolution. If the evolution times reach the maximum, the flag signal $flag$ will be set to 0, otherwise it will be set to 1. When all the $flags$ are 0, the algorithm will terminate. And then each distributed controller controls the chiller it connected for energy saving according to the result the DCEDA calculated.

5. Case study

In this study, two well-known cases are tested for illustrating the ability of DCEDA in solving the OCL problem. In the first case, the multi-chiller system, composed of three 800 RT units, was originally proposed by Chang et al. [34] to test the optimization ability of GA. Later, case 1 was used successively in Refs. [2, 3, 15, 17, 23, 24, 35] to test the performance of DE, GM, PSO, IFA, DCSA, EMA, EIWO and VAFSA. The second

Table 1

Performance parameters of chillers in two case studies [2,3,14–17,23,24,33–35].

System	Chiller	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	Capacity/ RT
Case 1	1	100.95	818.61	−973.43	788.55	800
	2	66.598	606.34	−380.58	275.95	800
	3	130.09	304.5	14.377	99.8	800
Case 2	1	399.345	−122.12	770.46	0	1280
	2	287.116	80.04	700.48	0	1280
	3	−120.505	1525.99	−502.14	0	1280
	4	−19.121	898.76	−98.15	0	1280
	5	−95.029	1202.39	−352.16	0	1250
	6	191.750	224.86	524.04	0	1250

subject is a semiconductor plant in Hsinchu Science-based Park [2,14–17,23,24,33,35], Taiwan, and the multi-chiller system is composed of four 1280 RT units and two 1250 RT units. It was also used to verify the ability of relevant optimization algorithms in solving the OCL problem. The values of the performance coefficients of chillers in two cases are shown in Table 1.

These cases represent two popular types of multi-chiller system. In case 1, the capacities and the types of chiller are all the same, but the characteristic curves are not identical as a result of the long running and the differences of chilled temperatures and flow rates between chillers coming from the system design structure. While case 2 is a typical multi-chiller system including chillers with different refrigeration capacity, but their performance curves are also different.

6. Experiment result and analysis

6.1. Parameter analysis

In order to find the best parameter values of DCEDA in solving the OCL problem, a lot of computational experiments with different parameter settings are performed. Case 1 with the system cooling load 2160 and 960 RT, case 2 with the system cooling load 6858 and 5334 RT, are selected as the experimental examples. Considering that the energy consumption of the multi-chiller system is accurate to 0.01, in order to make the deviation between the cooling capacity generated by the multi-chiller system and the system cooling load requirement less than 0.001, let $penal = 10000$.

Combined with the above conditions, all optimization results under different parameter settings are obtained based on 50 independent runs. The parameter values of DCEDA, minimum and mean value of optimization results are shown in Table 2. Experiment results indicate that in the same experimental example, the optimization results of different parameter settings are almost equal, and the minimum and mean optimization results obtained by the same parameter setting are also nearly identical. These mean that DCEDA has good robustness and strong stability. Furthermore, it can be found from the same experimental

example that the mean value of optimization results obtained by the parameter setting shown in boldface is less than that obtained by other parameter settings, and the minimum value of optimization results is also relatively better. In addition, the difference between the minimum and mean value of the optimization results obtained by the parameter setting shown in boldface is also smaller than that of other parameter settings. Therefore, from the above parameter analysis experiments, it can be concluded that the parameter setting shown in boldface has the best performance in solving the OCL problem, in which the population size $N = 80$, the new population multiplication $K = 1.5$, the number of better individuals $M = 5$ and the minimum variation step size $C_{min} = 0.2$. Then the best parameter setting of DCEDA found in the above parameter analysis experiments are applied to the following experiments.

6.2. Results and discussion

In the experiments, the parameter values of DCEDA used in two cases study are shown in Table 3. The optimization results of DCEDA are compared with those of GA [33,34], PSO [35,36] and DCSA [17], as shown in Tables 4 and 5. All the optimization results are obtained in 50 independent runs. The power energy saved by DCEDA compared to other algorithms are shown in boldface.

In case 1, compared with GA, DCEDA can save 2.82–149.93 kW power energy in each load demand, as shown in Table 4. Moreover, the lower the load demand is, the more power energy DCEDA can save than GA. And when the load demand is lower than 1440 RT, DCEDA can save more than 100 kW power energy than GA. In addition, the optimization results of DCEDA are equal to those of PSO and DCSA in each load demand, and the operation strategies obtained are also almost consistent.

In case 2, compared with GA and PSO, DCEDA can save 0.95–159.74 kW power energy in each load demand, and the lower the load demand is, the more obvious the energy saving effect of DCEDA is, as shown in Table 5. In addition, when the load demand is greater than or equal to 6096 RT, the operation strategy obtained by DCEDA is basically consistent with that of DCSA, and the energy consumption is also equal. However, when load demand is lower than 6096 RT, DCEDA has a higher power energy than DCSA. After simple analysis, it can be found that when the load is 5717 RT and 5334 RT, the PLR of No.3 chiller is less than 0.3 in the optimization results obtained by DCSA. According to the calculation, the power energy of No.3 chiller is −120.50 kW and

Table 3

Control parameters of the DCEDA in two cases study.

Symbol	Meaning	Value
<i>penal</i>	Penalty factor	10000
β	The parameter of one-dimensional logistic mapping	4
<i>N</i>	Number of population size	80
<i>K</i>	Newborn population multiplication	1.5
<i>M</i>	Number of better individuals	5
<i>C_{min}</i>	Minimum variation step size	0.2
<i>G_{max}</i>	Maximum number of iterations	50

Table 2

The optimum results under different parameter settings for four examples.

N	K	M	<i>C_{min}</i>	Case1(2160RT)		Case1(960RT)		Case2(6858RT)		Case2(5334RT)	
				Min	Mean	Min	Mean	Min	Mean	Min	Mean
60	1.5	5	0.2	1583.85	1585.12	692.25	692.84	4738.59	4739.01	3546.44	3549.13
80	1.5	5	0.2	1583.81	1583.98	692.25	692.39	4738.58	4738.66	3546.44	3547.09
100	1.5	5	0.2	1583.86	1584.84	692.25	692.52	4738.58	4738.82	3546.44	3548.80
80	1.2	5	0.2	1583.84	1584.97	692.25	693.22	4738.58	4739.27	3546.44	3549.06
80	1.8	5	0.2	1583.81	1585.32	692.26	692.54	4738.58	4738.72	3546.44	3548.65
80	1.5	4	0.2	1583.81	1585.17	693.27	692.79	4738.60	4739.07	3546.44	3548.97
80	1.5	6	0.2	1583.82	1584.91	692.26	692.77	4738.58	4738.70	3546.44	3548.32
80	1.5	5	0.1	1583.81	1584.73	692.25	692.63	4738.59	4738.95	3546.44	3548.72
80	1.5	5	0.3	1583.83	1585.13	692.26	693.62	4738.62	4739.00	3546.44	3549.38

Table 4

Comparison of minimal power consumption obtained by GA, PSO, DCSA and DCEDA in case 1.

Load demand	Chiller code	GA [34]		PSO [35]		DCSA [17]		DCEDA		Energy saving/kW		
		PLR	P/kW	PLR	P/kW	PLR	P/kW	PLR	P/kW	D-A	D-B	D-C
2160 (90%)	1	0.8050	1590.96	0.7300	1583.81	0.7253	1583.81	0.7265	1583.81	-7.15	0.00	0.00
	2	0.9323		0.9700		0.9747		0.9735				
	3	0.9631		1.0000		1.0000		1.0000				
1920 (80%)	1	0.7017	1406.02	0.6600	1403.20	0.6591	1403.20	0.6609	1403.20	-2.82	0.00	0.00
	2	0.7954		0.8600		0.8585		0.8557				
	3	0.9035		0.8800		0.8825		0.8834				
1680 (70%)	1	0.6900	1250.06	0.6000	1244.32	0.6000	1244.32	0.5942	1244.32	-5.74	0.00	0.00
	2	0.6784		0.7400		0.7400		0.7455				
	3	0.7318		0.7600		0.7600		0.7603				
1440 (60%)	1	0.5217	1107.75	0.0000	993.60	0.0000	993.60	0.0000	993.60	-114.15	0.00	0.00
	2	0.7407		0.8900		0.8963		0.8858				
	3	0.5381		0.9100		0.9037		0.9142				
1200 (50%)	1	0.4882	971.21	0.0000	832.33	0.0000	832.33	0.0000	832.33	-138.88	0.00	0.00
	2	0.4437		0.7400		0.7430		0.7425				
	3	0.5682		0.7600		0.7570		0.7575				
960 (40%)	1	0.3055	842.18	0.0000	692.25	0.0000	692.25	0.0000	692.25	-149.93	0.00	0.00
	2	0.3185		0.5700		0.5368		0.5683				
	3	0.5764		0.6300		0.6632		0.6317				

Table 5

Comparison of minimal power consumption obtained by GA, PSO, DCSA and DCEDA in case 2.

Load demand	Chiller code	GA [33]		PSO [36]		DCSA [17]		DCEDA		Energy saving/kW		
		PLR	P/kW	PLR	P/kW	PLR	P/kW	PLR	P/kW	D-A	D-B	D-C
6858 (90%)	1	0.7052	4766.33	0.8026	4739.53	0.8127	4738.58	0.8126	4738.58	-27.75	-0.95	0.00
	2	0.7693		0.7799		0.7496		0.7489				
	3	0.9868		0.9996		1.0000		1.0000				
	4	0.9868		0.9998		1.0000		1.0000				
	5	0.9794		0.9999		1.0000		1.0000				
	6	0.8842		0.8183		0.8386		0.8395				
6477 (85%)	1	0.6207	4459.16	0.7606	4423.04	0.7277	4421.65	0.7280	4421.65	-37.51	-1.39	0.00
	2	0.7742		0.6555		0.6561		0.6564				
	3	0.9927		1.0000		1.0000		1.0000				
	4	0.9589		1.0000		1.0000		1.0000				
	5	0.9956		1.0000		1.0000		1.0000				
	6	0.7595		0.6835		0.7165		0.7160				
6096 (80%)	1	0.8099	4185.87	0.6591	4147.69	0.6427	4143.71	0.6431	4143.71	-42.16	-3.98	0.00
	2	0.5474		0.5798		0.5626		0.5622				
	3	0.9878		0.9991		1.0000		1.0000				
	4	0.9624		0.9979		1.0000		1.0000				
	5	0.9897		0.9921		1.0000		1.0000				
	6	0.5029		0.5710		0.5945		0.5946				
5717 (75%)	1	0.5797	3940.60	0.7713	3921.07	0.8437	3840.06	0.0000	3843.07	-97.53	-78.00	3.01
	2	0.5621		0.7177		0.7838		0.7144				
	3	0.9428		0.3000		0.000001		1.0000				
	4	0.7908		0.9991		1.0000		1.0000				
	5	0.9951		1.0000		1.0000		1.0000				
	6	0.6339		0.7187		0.8830		0.7941				
5334 (70%)	1	0.5831	3706.22	0.6418	3642.55	0.7500	3507.27	0.0000	3546.48	-159.74	-96.07	39.21
	2	0.5767		0.6621		0.6825		0.5831				
	3	0.5230		0.3301		0.000012		1.0000				
	4	0.9497		0.9906		1.0000		1.0000				
	5	0.9521		0.9990		1.0000		1.0000				
	6	0.6207		0.5806		0.7764		0.6221				

-120.49 kW when the PLR is 0.000001 and 0.000012 respectively, which is impossible in practical application. Therefore, although the power energy calculated by DCSA under these two load demands are lower than that of DCEDA, they have no practical significance. In

conclusion, we still believe that the optimization results obtained by DCEDA are the same as those obtained by DCSA.

In two cases, the convergence curves are shown in Figs. 4–5. Overall, the fitness curves show an upward trend with the increase of the number

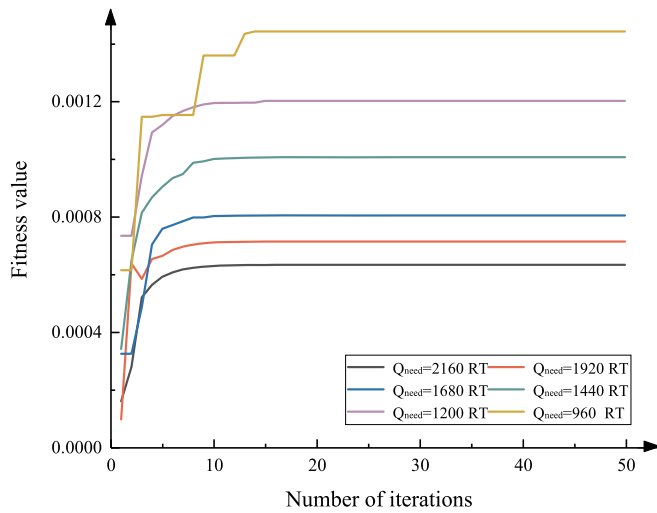


Fig. 4. Convergence curve of DCEDA in case 1.

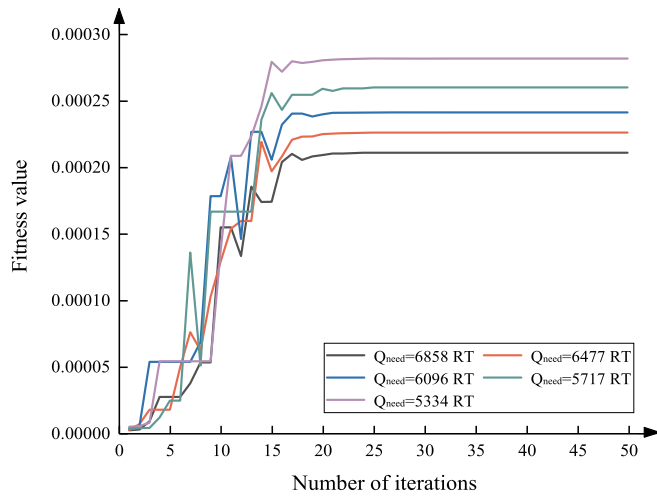


Fig. 5. Convergence curve of DCEDA in case 2.

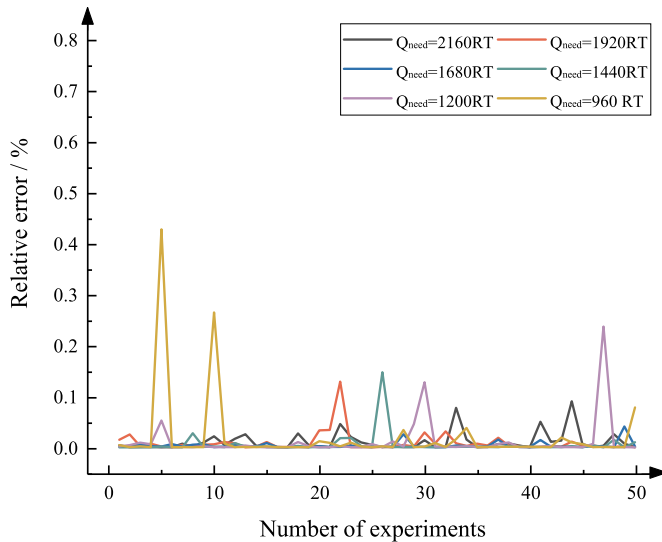


Fig. 6. Test results of DCEDA in case 1.

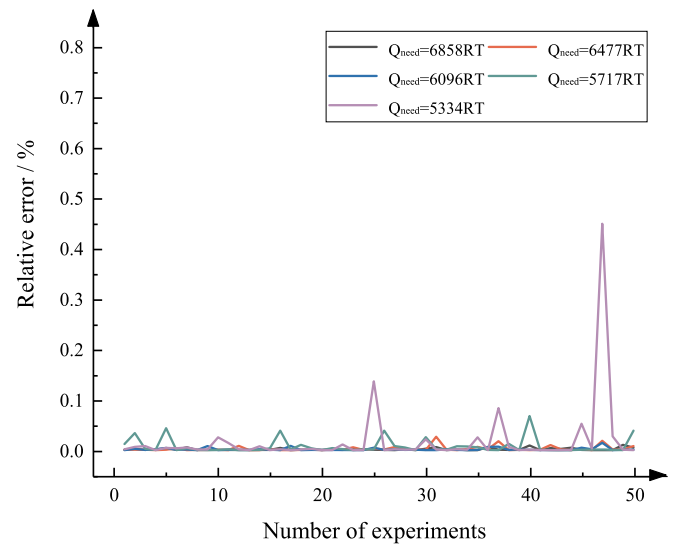


Fig. 7. Test results of DCEDA in case 2.

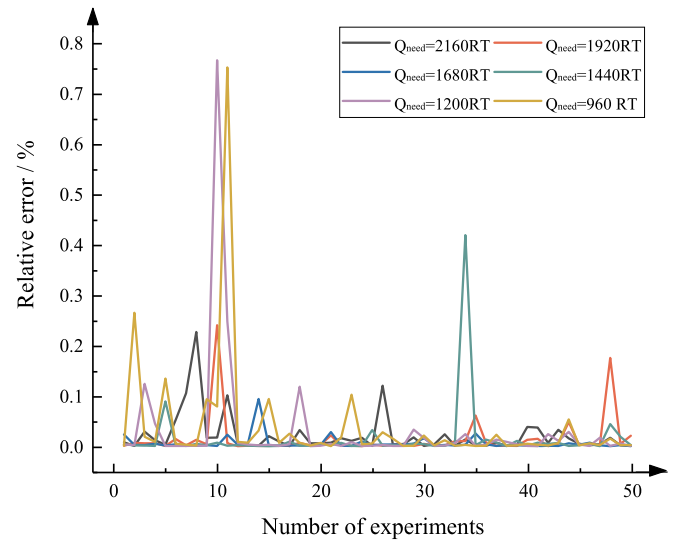


Fig. 8. Test results of DCSA in case 1.

of iterations. In case 1, the optimization results have been obtained after 15 generations of algorithm evolution, and in case 2, the algorithm has completed the evolution around 25 generations. Therefore, DCEDA is well convergent.

According to the above experimental results, the following three rules can be summarized. Firstly, the energy consumptions of operation strategies obtained by DCEDA in both cases are all lower than those obtained by GA. Secondly, the results of DCEDA are consistent with those of PSO in case 1, and are better than those of PSO in case 2. Finally, the operation strategies achieved by DCEDA are almost consistent with those achieved by DCSA except that the load demand is less than 6096RT in case 2. Therefore, it can be proved that DCEDA has better performance than GA and PSO in solving the OCL problem. Furthermore, to verify the superiority of DCEDA, the stability of DCEDA and DCSA are compared. The relative errors between the test results and the optimal value are shown in Figs. 6–9. The minimum, mean, maximum and the standard deviation of the optimization results in each load demand are shown in Tables 6 and 7. The parameter values of DCSA refer to the setting in Refs. [17].

In the further study of case 1, compared to DCSA, the relative errors

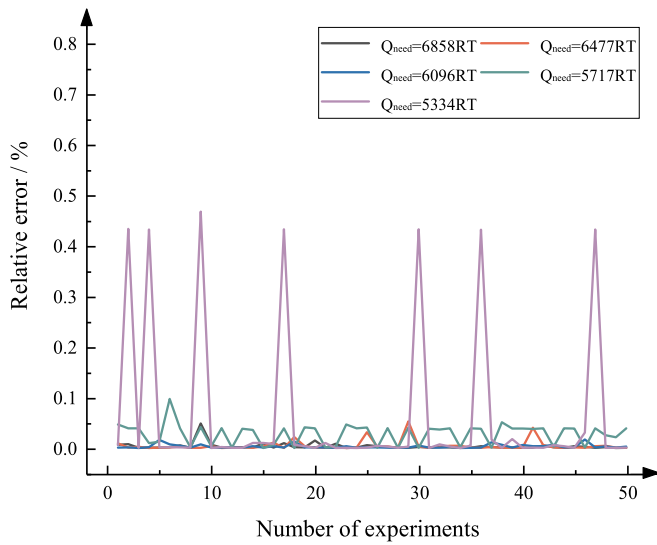


Fig. 9. Test results of DCSA in case 2.

Table 6

The comparison of optimum results of case 1 of DCSA and DCEDA.

Optimization method	Load demand	Minimum	Mean	Maximum	Standard deviation
DCSA	2160 (90%)	1583.81	1584.14	1587.40	0.622
DCEDA	2160 (90%)	1583.81	1583.98	1585.24	0.295
DCSA	1920 (80%)	1403.20	1403.40	1406.57	0.583
DCEDA	1920 (80%)	1403.20	1403.32	1405.01	0.272
DCSA	1680 (70%)	1244.32	1244.40	1245.49	0.176
DCEDA	1680 (70%)	1244.32	1244.37	1244.83	0.087
DCSA	1440 (60%)	993.60	993.75	997.77	0.591
DCEDA	1440 (60%)	993.60	993.66	995.07	0.209
DCSA	1200 (50%)	832.33	832.58	838.71	0.941
DCEDA	1200 (50%)	832.33	832.42	834.30	0.316
DCSA	960 (40%)	692.25	692.51	697.46	0.775
DCEDA	960 (40%)	692.25	692.39	695.22	0.485

of DCEDA are obviously smaller in each load demand, and the maximum value is lower than 0.5%, as shown in Figs. 6 and 8. In case 2, the results obtained by DCEDA are significantly better than those obtained by DCSA as the load is lower than 6096 RT, and the relative errors obtained by two algorithms are all close to 0 when the load is greater than 6096 RT, as shown in Figs. 7 and 9. In addition, the relative error curves obtained by DCEDA for 50 runs in each load demand are all flatter than those obtained by DCSA. In addition, as can be seen from Tables 6 and 7, the minimum of the results calculated by DCEDA and DCSA are all equal in each load demand. However, the mean value, maximum value and the standard deviation of the results obtained by DCEDA are all lower than those obtained by DCSA. What's more, except that the load demand is 5334 RT in case 2, the difference between the maximum and minimum energy consumption of the operation strategies obtained by DCEDA are all less than 3 kW. And the standard deviations of the optimization results of 50 independent runs are less than 1 kW. Therefore, DCEDA is superior to DCSA in algorithm stability.

Table 7

The comparison of optimum results of case 2 of DCSA and DCEDA.

Optimization method	Load demand	Minimum	Mean	Maximum	Standard deviation
DCSA	6858 (90%)	4738.58	4738.73	4740.88	0.340
DCEDA	6858 (90%)	4738.58	4738.66	4739.08	0.113
DCSA	6477 (85%)	4421.65	4421.85	4423.97	0.448
DCEDA	6477 (85%)	4421.65	4421.78	4422.83	0.232
DCSA	6096 (80%)	4143.71	4143.82	4144.40	0.160
DCEDA	6096 (80%)	4143.71	4143.78	4144.31	0.116
DCSA	5717 (75%)	3842.55	3843.61	3846.29	0.776
DCEDA	5717 (75%)	3842.55	3842.85	3845.16	0.557
DCSA	5334 (70%)	3546.44	3548.72	3563.04	5.352
DCEDA	5334 (70%)	3546.44	3547.09	3562.39	2.338

To sum up, from the aspects of energy saving, robustness, convergence and stability, DCEDA with the given parameter values based on the distributed framework can solve the OCL problem with a better performance than GA, PSO and DCSA.

7. Conclusion and future work

Aiming at the deficiency of centralized framework, a distributed framework for HVAC control is introduced and discussed. In the distributed framework, each device is upgraded as an agent by installing with a distributed controller and the system integrator only needs to connect the various agents through communication according to the actual physical topology to form the control network. The control network is a multi-agent network in which every agent has equal status. Besides, the algorithms are identical within the same type of device and are portable rather than developed case-by-case. In the optimization process, agents execute the algorithm and negotiate with neighbors to achieve the optimized results. Therefore, this control framework is simple in site construction and more in accord with the development of HVAC control system and even the intelligent building control system.

Furthermore, based on the distributed framework, a distributed optimization algorithm which combined with EDA and the chaotic mutation operator, is proposed to solve the OCL problem. The minimum energy consumption of chillers is considered as the optimization objective, the terminal cooling load demand is selected as the constraint, and the PLR of each chiller is as the optimization parameter. To verify the efficiency and effectiveness of the proposed algorithm, two well-known instances based on the OCL problem are tested and the results are compared with other recently published algorithms. The results of many computational experiments with different parameter settings show that DCEDA has good robustness and strong stability in solving the OCL problem. The experimental results show that DCEDA has better robustness, convergence and stability than other algorithms, and is a very effective energy saving algorithm in HVAC system. What's more, DCEDA is a distributed optimization algorithm that can run on the distributed control framework. It can be concluded that DCEDA is an effective and efficient distributed algorithm to solve the OCL problem, which can also be used to solve other optimal problems based on the distributed control framework.

In the future work, we will focus on the multi-objective DCEDA and its application in the multi-objective OCL problem based on the distributed control framework.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled “Optimal Chiller Loading in HVAC System Using a Novel Algorithm Based on the Distributed Framework”.

Acknowledgement

This paper is supported by Key R&D Projects in Shaanxi Province (2017ZDCXL-SF-03-02); National Key Research and Development Project of China, entitled New generation intelligent building platform techniques (No.2017YFC0704100); Shaanxi Provincial Science and Technology Department Special Research Project (2017JM6106); Brain Research Fund of the University (JC1706).

References

- [1] F.W. Yu, K.T. Chan, Energy signatures for assessing the energy performance of chillers, *Energy Build.* 37 (2005) 739–746.
- [2] Z.X. Zheng, J.Q. Li, Optimal chiller loading by improved invasive weed optimization algorithm for reducing energy consumption, *Energy Build.* 161 (2018) 80–88.
- [3] W.S. Lee, Y.T. Chen, Y.C. Kao, Optimal chiller loading by differential evolution algorithm for reducing energy consumption, *Energy Build.* 43 (2) (2011) 599–604.
- [4] T. Yang, X.L. Yi, J.F. Wu, Y. Yuan, D. Wu, A survey of distributed optimization, *Annu. Rev. Contr.* 47 (2019) 278–305.
- [5] P. Yi, Y.G. Hong, F. Liu, Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems, *Automatica* 74 (2016) 259–269.
- [6] S. Muelas, A. Mendiburu, A. LaTorre, J.-M. Peña, Distributed Estimation of Distribution Algorithms for continuous optimization: how does the exchanged information influence their behavior? *Inf. Sci.* 268 (2014) 231–254.
- [7] Y. Sun, G.G. Yen, Z. Yi, Reference line-based estimation of distribution algorithm for many-objective optimization, *Knowl. Based Syst.* 132 (2017) 129–143.
- [8] S. Gao, C.W. de Silva, Estimation distribution algorithms on constrained optimization problems, *Appl. Math. Comput.* 339 (2018) 323–345.
- [9] X. Wang, K. Xing, X. Li, J. Luo, An estimation of distribution algorithm for scheduling problem of flexible manufacturing systems using Petri nets, *Appl. Math. Model.* 55 (2018) 776–788.
- [10] M. Jiang, L. Qiu, Z. Huang, G.G. Yen, Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation, *Inf. Sci.* 435 (2018) 203–223.
- [11] Z. Wang, M. Gong, Dynamic deployment optimization of near space communication system using a novel estimation of distribution algorithm, *Appl. Soft Comput. J.* 78 (2019) 569–582.
- [12] Y.C. Chang, A novel energy conservation method-optimal chiller loading, *Electr. Power Syst. Res.* 69 (2) (2004) 221–226.
- [13] Y.C. Chang, T.S. Chan, W.S. Lee, Economic dispatch of chiller plant by gradient method for saving energy, *Appl. Energy* 87 (4) (2010) 1096–1101.
- [14] Z.W. Geem, Solution quality improvement in chiller loading optimization, *Appl. Therm. Eng.* 31 (10) (2011) 1848–1851.
- [15] L.d.S. Coelho, V.C. Mariani, Improved firefly algorithm approach applied to chiller loading for energy conservation, *Energy Build.* 59 (2013) 273–278.
- [16] M.H. Sulaiman, H. Ibrahim, H. Daniyal, M.R. Mohamed, A new swarm intelligence approach for optimal chiller loading for energy conservation, *Procedia - Soc. Behav. Sci.* 129 (2014) 483–488.
- [17] L.d.S. Coelho, C.E. Klein, S.L. Sabat, V.C. Mariani, Optimal chiller loading for energy conservation using a new differential cuckoo search approach, *Energy* 75 (2014) 237–243.
- [18] I.A. Mohamed, M. Kowsalya, Optimal size and siting of multiple distributed generators in distribution system using bacterial foraging optimization, *Swarm Evol. Comput.* 15 (2014) 58–65.
- [19] Y. Yang, B. Wei, H. Liu, Y.Y. Zhang, J. Zhao, E. Manla, Chaos firefly algorithm with self-adaptation mutation mechanism for solving large-scale economic dispatch with valve-point effects and multiple fuel options, *IEEE Access* 6 (2018) 45907–45922.
- [20] J. Pierezan, L.d.S. Coelho, Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems, 2018, IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, 2018, pp. 1–8.
- [21] H. Shayanfar, F.S. Gharehchopogh, Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems, *Appl. Soft Comput.* 71 (2018) 728–746.
- [22] N.A. Kallioras, N.D. Lagaros, D.N. Avtzis, Pity beetle algorithm - a new metaheuristic inspired by the behavior of bark beetles, *Adv. Eng. Software* 121 (2018) 147–166.
- [23] F. Sohrabi, M. Nazari-Heris, B. Mohammadi-Ivatloo, S. Asadi, Optimal chiller loading for saving energy by exchange market algorithm, *Energy Build.* 169 (2018) 245–253.
- [24] Z.X. Zheng, J.Q. Li, P.Y. Duan, Optimal chiller loading by improved artificial fish swarm algorithm for energy saving, *Math. Comput. Simulat.* 155 (2019) 227–243.
- [25] M.A. Latif, M.P. Saka, Optimum design of tied-arch bridges under code requirements using enhanced artificial bee colony algorithm, *Adv. Eng. Software* 135 (2019) 102685.
- [26] M. Najimi, N. Ghafouri, M. Nikoo, Modeling chloride penetration in self-consolidating concrete using artificial neural network combined with artificial bee colony algorithm, *J. Build. Eng.* 22 (2019) 216–226.
- [27] A. Javidi, E. Salajegheh, J. Salajegheh, Enhanced crow search algorithm for optimum design of structures, *Appl. Soft Comput.* 77 (2019) 274–289.
- [28] M. Petrović, Z. Miljković, A. Jokić, A novel methodology for optimal single mobile robot scheduling using whale optimization algorithm, *Appl. Soft Comput. J.* 81 (2019) 105520.
- [29] Q. Huang, W. Song, A land-use spatial optimum allocation model coupling a multi-agent system with the shuffled frog leaping algorithm, *Comput. Environ. Urban Syst.* 77 (2019) 101360.
- [30] M. Misaghi, M. Yaghoobi, Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller, *J. Comput. Des. Eng.* 6 (3) (2019) 284–295.
- [31] X. Tian, J. Li, A novel improved fruit fly optimization algorithm for aerodynamic shape design optimization, *Knowl. Based Syst.* 179 (2019) 77–91.
- [32] T.T. Nguyen, A high performance social spider optimization algorithm for optimal power flow solution with single objective optimization, *Energy* 171 (2019) 218–240.
- [33] Y.C. Chang, Genetic algorithm based optimal chiller loading for energy conservation, *Appl. Therm. Eng.* 25 (17) (2005) 2800–2815.
- [34] Y.C. Chang, J.K. Lin, M.H. Chuang, Optimal chiller loading by genetic algorithm for reducing energy consumption, *Energy Build.* 37 (2) (2005) 147–155.
- [35] W.S. Lee, L.C. Lin, Optimal chiller loading by particle swarm algorithm for reducing energy consumption, *Appl. Therm. Eng.* 29 (8) (2009) 1730–1734.
- [36] A.J. Ardakani, F.F. Ardakani, S.H. Hosseini, A novel approach for optimal chiller loading using particle swarm optimization, *Energy Build.* 40 (2008) 2177–2187.
- [37] C.L. Chen, Y.C. Chang, T.S. Chan, Applying smart models for energy saving in optimal chiller loading, *Energy Build.* 68 (2014) 364–371.
- [38] R.V. Rao, Optimization of multiple chiller systems using TLBO algorithm, *Teaching Learning Based Optimization Algorithm*, Springer, 2016, pp. 115–128.
- [39] P.Y. Duan, J.Q. Li, Y. Wang, H.Y. Sang, B.X. Jia, Solving chiller loading optimization problems using an improved teaching-learning-based optimization algorithm, *Optim. Control Appl. Methods* (2017) 1–13.
- [40] H. Teimourzadeh, F. Jabari, B. Mohammadi-Ivatloo, An augmented group search optimization algorithm for optimal cooling-load dispatch in multi-chiller plants, *Comput. Electr. Eng.* (2019) 1–12.
- [41] D. Jakovetić, J. Xavier, J.M.F. Moura, Fast distributed gradient methods, *IEEE Trans. Autom. Control* 59 (5) (2014) 1131–1146.
- [42] S.G.M. Rokni, M. Radmehr, A. Zakariazadeh, Optimum energy resource scheduling in a microgrid using a distributed algorithm framework, *Sustain. Cities Soc.* 37 (2018) 222–231.
- [43] C.K. Das, O. Bass, G. Kothapalli, T.S. Mahmoud, D. Habibi, Optimal placement of distributed energy storage systems in distribution networks using artificial bee colony algorithm, *Appl. Energy* 232 (2018) 212–228.
- [44] Y. Zheng, L. Jin, L. Gao, K. Li, Y. Wang, Development of a distributed cooperative vehicles control algorithm based on V2V communication, *Procedia Eng.* 137 (2016) 649–658.
- [45] Y. Jiang, M. Zanon, R. Hult, B. Houska, Distributed algorithm for optimal vehicle coordination at traffic intersections, *IFAC PapersOnLine* 50 (1) (2017) 11577–11582.
- [46] S. Triguia, A. Koubaa, O. Cheikhrouhou, H. Youssef, H. Bennaïce, M.F. Sriti, A distributed market-based algorithm for the multi-robot assignment problem, *Procedia Comput. Sci.* 32 (2014) 1108–1114.
- [47] L. Mark, Berliner, statistics, probability and chaos, *Stat. Sci.* 7 (1) (1992) 69–90.
- [48] Q. Lü, G. Shen, R. Yu, A chaotic approach to maintain the population diversity of genetic algorithm in network training, *Comput. Biol. Chem.* 27 (2003) 363–371.
- [49] Y.H. Cheng, X.S. Wang, M.L. Hao, An estimation of distribution algorithm with diversity preservation, *Acta Electron. Sin.* 38 (3) (2010) 591–597.