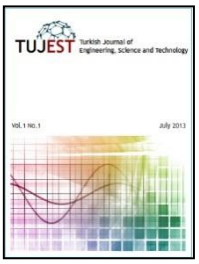
Turkish Journal of Engineering, Science and Technology 01 (2015) 47-66

TUJEST

Turkish Journal of Engineering, Science and Technology

journal homepage: www.tujest.com



Artificial Cooperative Search Algorithm for Optimal Loading Of Multi-Chiller Systems

Oğuz Emrah Turgut[a,] *, Mustafa Asker [b], Mustafa Turhan Çoban [a]

[a]*Department of Mechanical Engineering, Ege University, İzmir, Turkey*
[b]*Department of Mechanical Engineering Department, Adnan Menderes University, İzmir, Turkey*

Article Info

Article history:
Received Nov 04, 2014
Accepted Feb 13, 2015
Available online Apr 13, 2015

Keywords:
Artificial Cooperative Search,
Decoupled System,
Energy Conservation,
Optimal Chiller Loading

Abstract

This study proposes Artificial Cooperative Search (ACS) algorithm to solve Optimal Chiller Loading problems and to cope with the difficulties of traditional optimization methods. ACS is a swarm based meta-heuristic algorithm which mimics the migration behavior of two artificial super-organisms as they interact each other to find global best solution of the corresponding problem. Partial Load Ratio (PLR) is selected as a decision variable to be optimized as main objective of this problem is to minimize total energy consumption of system. Case studies adopted from literature are utilized for testing the performance of ACS algorithm. Results of ACS are compared with those obtained from Artificial Bee Colony, Quantum behaved Particle Swarm Optimization, Cuckoo Search, and literature approaches. Comparisons indicate that ACS algorithm surpasses other optimization methods in terms of solution efficiency and accuracy.

1. Introduction

Chiller systems are generally used in various air conditioning applications. These types of systems require considerable electrical energy for their operations imposing huge loads and demands on electrical energy supply specially during operations in hot weathers, leading to undesirable peak loads. Thus, the optimization of chiller design and performance has become of an utmost importance and received a great deal of interest and attention because of its role in reducing energy consumptions and operational costs.

Several authors have investigated the optimal chiller loading problem by using various types of optimization methods. Chang [1] used Lagrangian method to solve the optimal chiller loading to cope with difficulties of conventional methods. The coefficient of performance (COP) of a chiller is chosen as the objective function for the reason of being a concave function. Chang et al.[2] and Chang [3] used general algorithm to solve optimal chiller loading (OCL) problem to overcome the deficiencies of the Lagrangian method as the system may not converge at low power demands. Chang et al.[4] and Chang [5] utilized Simulated Annealing to solve the optimal chiller loading

*Corresponding Author:
Oguz Emrah Turgut, e-Mail: oeturgut@hotmail.com

(OCL) problem to eliminate the limitation of Lagrangian method which cannot solve the OCL as the power consumption model or the kW–PLR (kilo Watt–partial load ratio) curves involve convex and non-convex functions.

Ardakani, Ardakani and Hosseinian [6] employed Continuous Genetic Algorithm (C-GA) and Particle Swarm Optimization (PSO) to solve optimal chiller loading (OCL) problem. These methods have major advantages such as fast convergence, escaping from trapping into local optimum and simple implementation to overcome deficiencies of other conventional optimization methods. They chose partial load ratio (PLR) of the chiller as a design variable to be optimized and consumption power of the chiller is considered as fitness function. They showed that both of these methods find the optimal solution while the equality constraint is exactly satisfied. Chang and Chen [7] employed the Hopfield Neural Network (HNN) to decide the chilled water supply temperatures which are used to solve the optimal chiller loading (OCL) problem.

Lee and Lin [8] applied Particle Swarm Optimization algorithm to minimize energy consumption of multi-chiller system. The objective function is to minimize the energy consumption and the decision variable is the partial loading ratio of each chiller. They showed that Particle Swarm Optimization algorithm outperforms Genetic Algorithm not only in overcoming the divergence problem of Lagrangian method occurring at low demands, but also in obtaining the minimum energy consumption solution. Lee et al. [9] employed Differential Evolution algorithm to solve the optimal chiller loading problem for reducing energy consumption. They showed that the proposed Differential Evolution algorithm can find the optimal solution as the Particle Swarm Optimization does but with better average solutions. In addition, it outperforms Genetic Algorithm in finding optimal solution and overcomes the divergence problem caused by the Lagrangian method occurs at low power demands. Beghi et al. [10] presented Particle Swarm Optimization (PSO) algorithm for optimal operation of multiple chiller systems. They solved Optimal Chiller Loading (OCL) and Optimal Chiller Sequencing (OCS) problems simultaneously in order to achieve optimal performance in terms of reducing both power consumption and operative costs, as well as granting good load tracking properties. They showed that PSO satisfactorily deals with such kind of nonlinear constrained optimization problem. Coelho and Mariani [11] proposed Improved Firefly Algorithm (IFA) based on Gaussian distribution function to the optimal chiller loading design. The objective function is to minimize the energy consumption and the optimized parameters are the partial loading ratios of each chiller. They acquired better results in comparison with other optimization methods.

The aim of this work is to minimize the energy consumption of decoupled chiller system with using ACS algorithm and to benchmark the performance of ACS algorithm with swarm based metaheuristic algorithms such as Artificial Bee Colony (ABC) [12], Quantum behaved Particle Swarm Optimization (QPSO) [13,14], Cuckoo Search (CS) [15] algorithms, and literature approaches by simulating three case studies. Remainder of the paper is constructed as follows. Section 2 gives the brief description of decoupled multi-chiller system and describes the objective function. Section 3 presents the description of Artificial Cooperative Search algorithm in details. Implementation of ACS algorithm on optimal chiller loading problem is explained in Section 4. Results of the case studies are presented and discussed in Section 5. Paper is concluded with remarkable comments in Section 6.

2. System description

A multiple chiller system has two or more chillers connected by parallel or series piping to a common distribution system. Multiple chillers have many advantages such as it offers operational flexibility, standby capacity, and less disruptive maintenance. The chillers can be sized to handle a base load and increments of a variable load to allow each chiller to operate at its most efficient point. A brief discussion related to typical chiller system is expressed in detail in ASHRAE Handbook [16]. Figure 1 illustrates decoupled chiller system consisting of multiple chillers.

Decoupled chiller system is designed specifically for variable-speed pumping. In the primary loop, fixed-speed pumps provide a relatively constant flow of water to the chillers. This design ensures good chiller performance and reduces the risk of freezing on evaporator tubes. The secondary loop incorporates one or more variable-speed pumps that are controlled to maintain chilled water loop differential pressure set point [17]. The multiple pumps are connected by a bypass pipe that connects the return and supply headers. Each chiller-pump combination operates independently from the remaining chillers. Capacity control is simplified and as if each chiller operated alone.

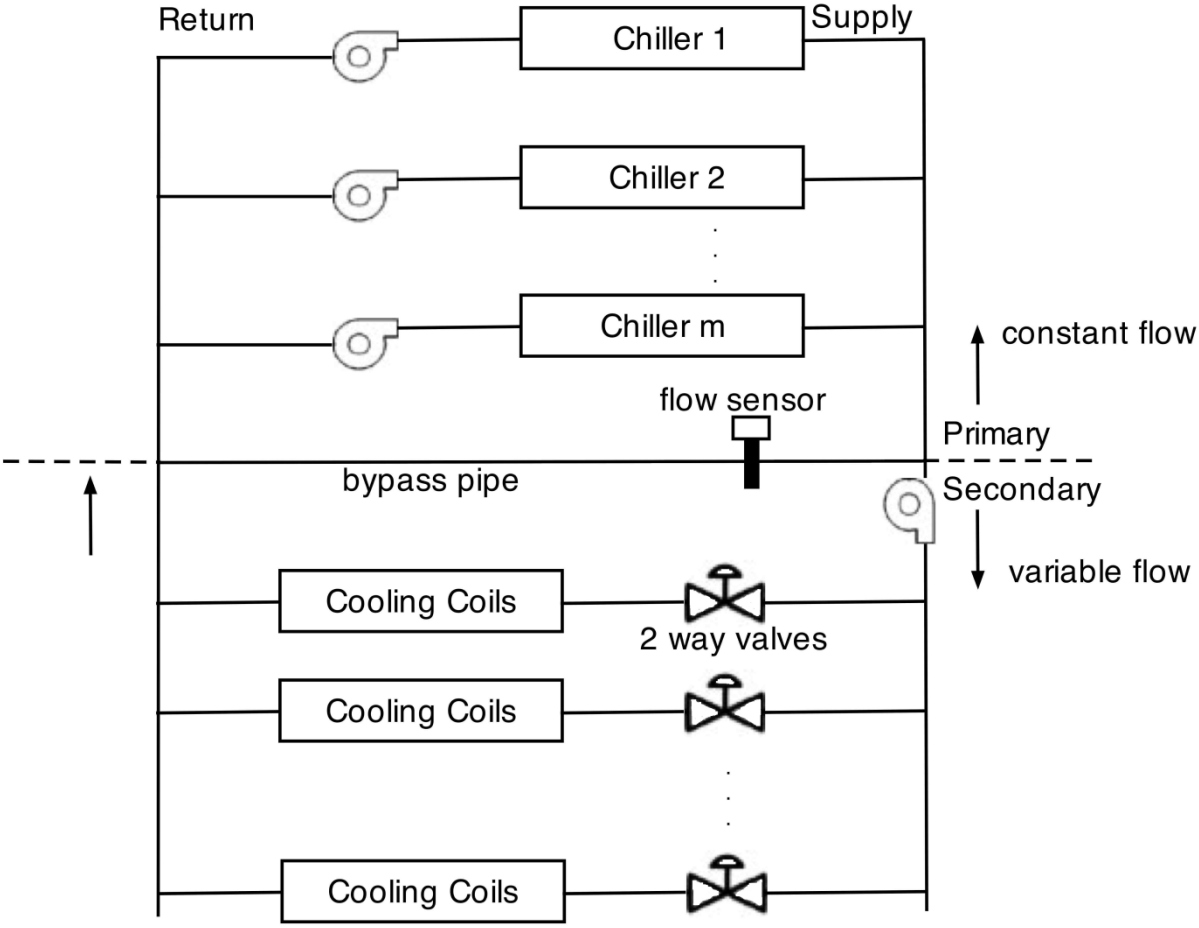


Figure 1. Multiple decoupled chiller system [16]

In a multiple chiller system with all-electric cooling, the best performance occurs when the sum of energy consumption of each chiller is minimized while the load demand is satisfied. The partial load ratio (PLR) is defined as the ratio of the chiller cooling load to the chiller power consumption [17]. The energy consumption is a convex function of its PLR in a given wet-bulb temperature. In this paper, the power consumption of a centrifugal chiller is expressed as [2,11].

$$P_i = a_i + b_i \times PLR_i + c_i \times PLR_i^2 \quad (1)$$

or

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

where a_i, b_i, c_i and d_i are the coefficients of interpolation for consumed power versus PLR of i^{th} chiller. The optimal chiller loading problem aims to find a set of chiller output which does not violate the operating limits while minimizing the objective function J given by

$$J = \sum_{i=1}^{m} P_i \quad (3)$$

The objective function J is the sum of consumed power by each chiller, where m refers to the total number of chillers and P_i is the consumed power by i^{th} chiller. Simultaneously, the balance equation must be satisfied. This constraint is stated as follows:

$$\sum_{i=1}^{m} PLR_i.RT_i = CL \quad (4)$$

where RT_i is the capacity of i^{th} chiller and CL is demanded cooling load. The constraint given by Equation (4) is a condition that must be satisfied in order for the design to be feasible. Another constraint is that partial load ratio of each demanded chiller should be higher than 0.3 in accordance with the supplier's recommendation, as shown in Equation (5).

$$1.0 \geq PLR_i \geq 0.3 \quad (5)$$

3. Artificial Cooperative Search

In order to solve complicated engineering design problems, many optimization algorithms have been developed for decades. These algorithms divide into two categories, namely, derivative based (analytical) optimization methods and heuristic methods. Analytical optimization techniques such as Steepest Descend and Newton based methods have better convergence characteristics than heuristic approaches, acquire the optimum with consistency and are more capable in fulfilling local search task however, they face with difficulties while solving real world design problems which are generally large scale and high dimensional. These types of algorithms suffer from computational requirements as well as the selection of the appropriate starting point. In addition, they cannot cope with nonlinearities and non-convexities of the objective function of the problem. These computational drawbacks have urged researchers to develop more efficient methods those aim to reach global optimum with minimum deficiency. A promising alternative for traditional algorithms can be given as a kind of nature inspired soft computing technique called metaheuristics.

Recently, metaheuristic algorithms are in demand for solving optimization problems since they are derivative free and do not need domain information of corresponding objective function. They offer advanced search strategy in optimization process with combining the randomness and the rules of the natural phenomena [18]. Owing to stochastic discrepancy, solutions obtained by these algorithms are not guaranteed to be global optimum. Nonetheless, they attain near global best solutions within a reasonable elapsed time.

Metaheuristic algorithms such as Genetic algorithms [19], Differential Evolution [20], Ant Colony Optimization [21], Simulated Annealing [22] and Particle Swarm Optimization [23] algorithms have been applied to numerous science and engineering design problems. There also exists Harmony Search [24], Artificial Bee Colony [12], Firefly [25] and Bat Algorithms [26] which are relatively new emerged and nature inspired optimization methods. Main drawback of these algorithms is that their convergence performance and solution success vary from problem to problem, that is, they are problem-dependent. Proper selection of the metaheuristic algorithm is a must to obtain global best solution of the mentioned optimization problem.

Artificial Cooperative Search (ACS) is a swarm intelligent (SI) based evolutionary algorithm for solving numerical optimization problems proposed by Civicioglu [27]. The algorithm rests on interaction between artificial superorganisms as they interact with each other and migrate to different feeding zones to find global minimum of a problem.

In nature, amount of food that can be discovered in a zone depends on yearly climate changes, because it's highly diverse. Therefore, many superorgranisms demonstrate some kind of seasonal migration behavior to find more suitable feeding zones. Also, it's known that, in nature, most species form superorganisms and split up to sub-groups (sub-superorganisms) prior to migration to find better feeding zones. Behavior of the superorganism is decided by the coordination of sub-groups. Two distinguishable behaviors of the superorganisms are interaction and explorer usage. Before migrating to a new area, first, superorganisms send an explorer to gather information about the possible migration area. Then, explorer shares the information with the superorganism to decide if the area is suitable for migration. Exploration process of the superorganisms continues even in migration. Interaction is another important behavior for living species. All superorganisms living in the same habitat, naturally interacts with each other. Parasite/host or predator/prey relationships may emerge in alturation, coextinction, coevolution or cooperation interactions between superorganisms.

In ACS algorithm, two superorganisms stated as α and β those contain random solutions of the problem, move to more suitable nesting and feeding zones. Each of the superorganism consists of N sub-superorganisms which

also refer to size of the population and each sub-superorganism consists of *D* members correspond to dimension of the problem. Also, the two superorganisms decide prey and predator sub-superorganisms. Predator sub-superorganism tracks the prey sub-superorganism while they move towards global minimum of the problem.

First, the initial values of the individuals of the two superorganisms are determined by using Equation (6).

$$\alpha_{i,j:g} = rnd.(up_j - low_j) + low_j$$
$$\beta_{i,j:g} = rnd.(up_j - low_j) + low_j \quad (6)$$

Where $i = 1,2,3,...,N$, $j = 1,2,3,...,D$ and $g = 1,2,3,...,\text{maxcycle}$. The *g* value represents the iteration number (generation) while *rnd* represents a random number chosen from a uniform distribution between [0,1]. up_j and low_j represents the upper and lower bounds of the search space for j^{th} dimension of the problem. Fitness values of the associated sub-superorganisms are calculated by using Equation (7).

$$y_{i;\alpha} = f(\alpha_i)$$
$$y_{i;\beta} = f(\beta_i) \quad (7)$$

Predator superorganism is decided by the rule in Algorithm 1.

Algorithm 1 Calculation of Predator individuals

if $rnd < rnd$
  Predator = α, $y_{\text{Predator}} = y_a$, key = 1
else
  Predator = β, $y_{\text{Predator}} = y_\beta$, key = 2
end

Prey superorganism is determined by the decision rule in Algorithm 2.

Algorithm 2 Calculation of Prey individuals

if $rnd < rnd$ Prey = α *else* Prey = β *end*
Prey = permute (Prey)

In Algorithm 2, *permute()* function randomly changes the places of row elements of prey individuals. Passive individuals are determined by the rule defined in Algorithm 3 and they are eliminated because only active individuals are permitted to migrate.

Algorithm 3 Calculation of passive individuals by binary valued integer map (*M*)

$M_{N\ x\ D} = 1$
for all elements in M
  if $rnd < (p \times rnd)$ *then* $M_{rndint(N),\ rndint(D)} = 0$ *end*
end
if $rnd < (p \times rnd)$ *then*
  for i=1 to N
    for j=1 to D
      if $rnd < (p \times rnd)$ *then* $M_{i,j} = 1$ *else* $M_{i,j} = 0$ *end*
    end
  end
end
for i=1 to N
  if $\sum_{j=1}^{D} M_i = D$ *then* $M_{i,rndint(D)} = 0$ *end*
end

In Algorithm 3, *rndint()* function generates random integers between selected interval by using gauss distribution and *p* represents probability of biological interaction. Biological interaction location between prey and predator individuals is calculated by using the following equation:

$$x = Predator + R\ x\ (Prey - Predator) \tag{8}$$

R, the variable that controls the speed of biological interaction, is calculated by using the following rule.

Algorithm 4 Decision rule to obtain scale factor (*R*)

if $rnd < rnd$ *then*
  $R = 4 \times rnd \times (rnd - rnd)$
else
  $R = \Gamma(4 \times rnd, 1)$
end

where Γ() represents the gamma distribution with a shape parameter of $4 \times rnd$ and a scale parameter of 1.0. Position updating by active individuals is shown in Algorithm 5.

Algorithm 5 Updating of biological interaction locations by active individuals

for i=1 to N
  for j=1 to D
    if $M_{i,j} > 0$ *then* $x_{i,j} = Predator_{i,j}$ *end*
  end
end

If biological interaction locations violate boundaries, new locations are generated according to the rule described in Algorithm 6.

Algorithm 6 Application of boundary control mechanism

```
for i=1 to N
    for j=1 to D
        if (x_{i,j} < low_j) ∨ (x_{i,j} > up_j) then
            x_{i,j} = low_j + rnd × (up_j − low_j)
        end
    end
end
```

Predator sub-superorganisms are compared with the biological interaction zones in terms of cost function value. If fitness value of the biological interaction zones are better, than updating of Predator individuals by applying the rule in Algorithm 7.

Algorithm 7 Updating of Predator sub-superorganism

```
for i=1 to N
    if f(x_i) < y_{i,Predator} then
        Predator_i = x_i
        y_{i,Predator} = f(x_i)
    end
end
```

New α and β superorganisms and their fitness values for next generations are decided by the rule in Algorithm 8, with the utilization of "key" parameter decided in Algorithm 1.

Algorithm 8 Determination of new sub organisms for next generations

```
if key=1 then
    α = Predator, y_α = y_Predator
else
    β = Predator, y_β = y_Predator
end
```

4. Implementation of Artificial Cooperative Search Algorithm on Optimal Chiller Loading Problem

ACS algorithm is a swarm-based algorithm which is constructed on the interaction of prey and predator individuals. ACS is based on the basics of selection, mutation and crossover strategies to find global minimum of the related problem. General steps of the implementation procedure of ACS algorithm on optimal chiller loading problem are given below.

Step 1: Main purpose of the OCL is to reduce the total power consumption of an HVAC system while satisfying demanded cooling load constraints. Therefore, at first, initialize the problem dimension according to the number of chiller that takes place in the operating system. Determine the population size, the maximum number of iteration

and the lower and upper bounds of the OCL problem. Then, initialize superorganism populations (*α and β*) with Equation (6) and calculate their fitness values (*J*) with the equation below

$$J = \sum_{i=1}^{m} P_i + PF \tag{9}$$

where P_i is the power load of the i^{th} chiller and m is the number of the chiller in the system as defined before. P_i can be calculated with either Equation (1) or Equation (2) according to the nature of the applied case study. ER value in Equation (9) represents the subjected constraint on the optimization problem. For a successful solution, all the imposed constraints should be satisfied accurately. In this context, power demand of the whole system should be equal to the total power capacity of the chillers. This behavior is characterized by the following equation given as

$$ER = \left| \sum_{i=1}^{m} \left(PLR_i \times RT_i \right) - CL \right| \tag{10}$$

ER function represents the absolute difference between the total power demand and total power output of the multi-chiller system. This value should be close enough to zero in order to satisfy the total power load with chiller capacity.

Penalty factor is another important issue in optimization process. By applying penalty factor properly, constrained optimization problem turns into unconstrained one. It is a problem dependent parameter which penalizes the unfeasible solutions of the problem. In this study, this parameter is set to 20.0 for each case. Therefore, Penalty Function (PF) can be represented as the following equation

$$PF = 20.0 \times ER \tag{11}$$

Considering the *PF*, objective function of the problem (fitness value) can be written in details as with the following equation

$$J = \sum_{i=1}^{m} P_i + 20 \times \left| \sum_{i=1}^{m} \left(PLR_i \times RT_i \right) - CL \right| \tag{12}$$

Step 2 Determine Predator individuals by using the rule in Algorithm 1.

Step 3 Calculate Prey individuals by applying Algorithm 2.

Step 4 Determine the scale factor (R) with Algorithm 4.

Step 5 Define passive individuals by using binary valued integer map as clarified in Algorithm 3.

Step 6 Determine biological interaction zones with Equation (8).

Step 7 Update biological interaction zones of active individuals by using Algorithm 5.

Step 8 Apply boundary control with Algorithm 6.

Step 9 Update Predator sub-superorganisms by applying Algorithm 7. Utilize Equation (12) to calculate fitness values

Step 10 Assess new superorganisms for next generations by applying Algorithm 8.

Step 11 Store the best solution and its corresponding fitness value.

Step 12 Repeat Step 2 to Step 11 until termination criteria is met.

5. Case studies and results

In order to assess the effectiveness of ACS algorithm on high dimensional real world optimization problem concerning the determination of the optimal loading of a multiple chiller system, three case studies taken from literature are successfully solved. Chiller data utilized in constructing the objective function of corresponding cases are given in Table 1. In addition, performance of swarm based algorithms such as Cuckoo Search (CS) [15], Quantum behaved Particle Swarm Optimization (QPSO) [13,14] and Artificial Bee Colony optimizations (ABC) [12] are tested and statistical results for these algorithms are compared with ACS algorithm. Algorithm parameters for each optimizer considering all case studies are shown in Table 2. Simulations are performed in Java™ executing Pentium Core i5 CPU @ 2.5 GHz and 6.0 GB RAM on personal computer. Due to stochastic discrepancy, 100 consecutive algorithm runs are performed for each case.

Table 1 Chillers data for first, second, and third cases

System		a_i	b_i	c_i	d_i	Capacity
Case 1	Chiller 1	399.345	-122.12	770.46	-	1280
	Chiller 2	287.116	80.04	700.48	-	1280
	Chiller 3	-120.505	1525.99	-502.14	-	1280
	Chiller 4	-19.121	898.76	-98.15	-	1280
	Chiller 5	-95.029	1202.39	-352.16	-	1250
	Chiller 6	191.750	224.86	524.04	-	1250
Case 2	Chiller 1	104.09	166.57	-430.13	512.53	450
	Chiller 2	-67.15	1177.79	2174.53	1456.53	450
	Chiller 3	384.71	-779.13	1151.42	-63.2	1000
	Chiller 4	541.63	413.48	-3626.5	4021.41	1000
Case 3	Chiller 1	100.95	818.61	-973.43	788.55	800
	Chiller 2	66.598	606.34	-380.58	275.95	800
	Chiller 3	130.09	304.5	14.377	99.8	800

Table 2 Assigned parameter values for each algorithm

Algorithms	Parameters	Case Study 1	Case Study 2	Case Study 3
ACS	Population size	20	20	20
	Maximum number of generation	8000	7500	7000
ABC	Population size	20	20	20
	Number of limited food source	2000	1500	1500
	Maximum number of cycle	8000	7500	7000
QPSO	Population size	20	20	20
	Maximum number of generation	8000	7500	7000
CS	Number of nests	100	90	80
	Discovery rate of alien eggs	0.25	0.25	0.25

5.1 Case study 1: Six Chillers

This case, which was firstly proposed by Chang [3], is based on the experimental data from the six chillers in a semiconductor factory located in Hsinchu Scientific Garden. Problem was previously solved with average loading method (AVL) [3] and Genetic algorithm (GA)[3], Simulated Annealing (SA) [5], Binary Genetic Algorithm (B-GA) [6], Continuous Genetic Algorithm (C-GA) [6], Particle Swarm Optimization (PSO) [6], Generalized reduced Gradient Method (GRG) [28] and Evolution Strategy technique(ES) [29]. Table 3 gives the comparison results of ACS algorithm with ABC, QPSO, CS and previous literature studies. Table 3 clearly shows that ACS algorithm surpasses other algorithms for all power load cases in terms of best solution. Table 4 reports the statistical results of ACS, ABC, QPSO and CS algorithm for different power loads. Table 4 clarifies that ACS algorithm outperforms other algorithms

considering minimum objective function value and standard deviation value. And also, it is observed that even the worst solution obtained by ACS is better than the best solution of other swarm based algorithms portrayed in Table 4 for all cases. Figure 2 plots the power savings for each load case when ACS algorithm is utilized instead of average power loading (AVL). As revealed in Figure 2, maximum saving (215 kW) is attained at the power load of 80%. CPU times for ACS, CS, ABC and QPSO algorithms considering all power load cases are given in Table 5. Average calculation times for ACS, ABC, QPSO, and CS are respectively 0.3880, 0.1328, 0.3284, 51.2428 seconds as given in Table 5.

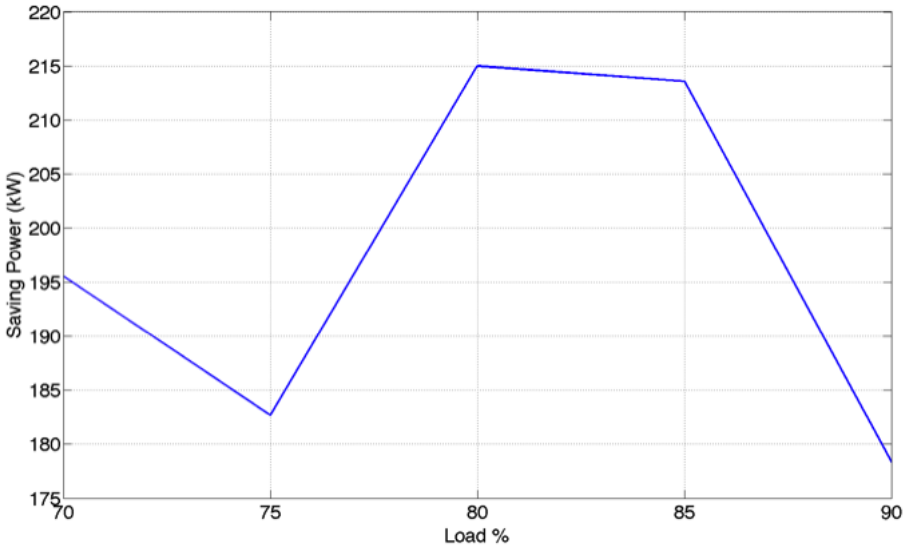


Figure 2. Power saving plot for case study 1

Table 3 Comparison of the best results obtained from different methods

RT	Chiller	AVL [3]		GA [3]		SA [5]		B-GA [6]		C-GA [6]		PSO [6]	
	i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i
6858(90%)	1	0.90000	913.51	0.7052	789.44	0.7789	771.68	0.8473	836.6782	0.8305	829.3918	0.8026	797.6788
	2	0.90000	921.54	0.7693	763.26	0.7587	751.04	0.7266	674.8695	0.7477	738.6028	0.7799	775.6981
	3	0.90000	846.15	0.9868	896.37	0.9791	892.24	0.9996	902.7402	1.0000	903.3450	0.9996	903.1638
	4	0.90000	710.26	0.9868	772.20	0.9781	766.03	0.9989	779.2732	0.9999	781.4819	0.9998	781.3991
	5	0.90000	701.87	0.9794	744.83	0.9820	746.11	0.9992	755.1275	1.0000	755.2010	0.9999	755.1979
	6	0.90000	818.60	0.8842	800.23	0.9265	849.88	0.8287	795.9624	0.8222	730.9420	0.8183	726.6468
	∑		4916.933		4766.33		4776.98		4744.6512		4738.9645		4739.7845
6477(85%)	1	0.85000	852.20	0.6207	620.38	0.8051	800.38	0.8261	824.2076	0.8068	802.3774	0.7606	752.2134
	2	0.85000	861.25	0.7742	768.94	0.6056	592.49	0.5672	557.8702	0.6588	643.9125	0.6555	640.5199
	3	0.85000	813.79	0.9927	899.51	0.9689	886.65	0.9985	902.5429	1.0000	903.3450	1.0000	903.3449
	4	0.85000	673.91	0.9589	752.45	0.9941	777.36	0.9715	761.4182	1.0000	781.4890	1.0000	781.4889
	5	0.85000	672.57	0.9956	753.00	0.9866	748.49	0.9972	753.8255	1.0000	755.2010	1.0000	755.2010
	6	0.85000	761.50	0.7595	664.82	0.7432	648.27	0.7404	645.4848	0.6327	543.8299	0.6835	590.2852
	∑		4635.215		4459.10		4453.64		4445.3493		4430.2444		4423.0534
6096(80%)	1	0.80000	794.74	0.8099	805.81	0.5635	575.21	0.7381	728.9208	0.6519	647.1984	0.6591	653.5696
	2	0.80000	799.46	0.5474	540.83	0.5743	564.15	0.4514	465.9823	0.6147	601.0118	0.5798	569.0161
	3	0.80000	778.92	0.9878	896.91	0.9675	885.84	0.9856	895.7077	0.9999	903.3449	0.9991	902.8647
	4	0.80000	637.07	0.9624	754.94	0.9798	767.27	0.9670	758.2284	0.9992	780.9059	0.9979	780.0799
	5	0.80000	641.50	0.9897	750.03	0.9845	747.41	0.9981	754.2742	0.9999	755.2001	0.9921	751.2365
	6	0.80000	707.02	0.5029	437.37	0.7338	638.91	0.6612	569.5019	0.5325	460.1566	0.5710	491.0385
	∑		4358.711		4185.88		4178.80		4172.6155		4147.8178		4147.8055
5717(75%)	1	0.75000	741.14	0.5797	587.47	0.6140	614.83	0.6139	614.7192	0.5962	600.4008	0.7713	763.4782
	2	0.75000	741.17	0.5621	553.43	0.4429	459.98	0.4953	498.6126	0.4685	478.3623	0.7177	705.3382
	3	0.75000	741.53	0.9428	871.86	0.9891	897.59	0.9988	902.7272	1.0000	903.3450	0.3000	292.0994
	4	0.75000	599.74	0.7908	630.24	0.8867	700.36	0.9413	739.9096	0.9999	781.4888	0.9991	780.8389
	5	0.75000	608.67	0.9951	752.75	0.9841	747.17	0.9999	755.1976	1.0000	755.2010	1.0000	755.2010
	6	0.75000	655.17	0.6339	544.86	0.5878	504.95	0.4511	399.8417	0.4353	388.9640	0.7187	624.0084
	∑		4087.419		3940.61		3925.16		3911.0079		3907.7607		3920.9642
5334(70%)	1	0.70000	691.39	0.5831	590.10	0.6265	625.28	0.6506	646.0258	0.6237	622.8603	0.6418	638.3097
	2	0.70000	686.38	0.5767	566.24	0.7403	730.12	0.5206	518.6970	0.4964	499.4574	0.6621	647.2355
	3	0.70000	701.64	0.5230	540.24	0.3093	303.40	0.9989	902.8016	0.9999	903.3442	0.3301	328.5020
	4	0.70000	561.92	0.9497	745.91	0.9546	749.39	0.5782	497.7299	0.5733	463.8435	0.9906	774.8633
	5	0.70000	574.09	0.9521	730.54	0.9511	730.03	0.9873	748.8425	1.0000	755.2010	0.9990	754.6915
	6	0.70000	605.93	0.6207	533.22	0.6250	536.96	0.4654	409.9351	0.5092	442.1532	0.5806	498.9765
	∑		3821.339		3706.24		3675.18		3694.0319		3686.8597		3642.5786

Table 3 Comparison of the best results obtained from different methods (continued)

RT	Chiller	GRG [28]		ES [29]		ABC		QPSO		CS		ACS	
	i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i
6858(90%)	1	0.8127	809.00	0.82	821.52	0.8279	826.371	0.8093	805.138	0.8130	810.460	0.8127	808.987
	2	0.7496	740.74	0.75	742.42	0.7339	723.207	0.7458	736.429	0.7493	740.425	0.7496	740.767
	3	1.0000	903.35	1.00	903.35	1.0000	903.345	0.9999	903.292	0.9999	903.344	0.9999	903.345
	4	1.0000	781.49	1.00	781.49	1.0000	781.489	0.9999	781.418	1.0000	781.489	0.9999	781.489
	5	1.0000	755.20	1.00	755.20	0.9999	755.195	0.9999	755.151	0.9999	755.200	0.9999	755.201
	6	0.8386	748.80	0.83	734.78	0.83903	749.324	0.8458	756.823	0.8385	748.789	0.8385	748.785
	∑		4738.58		4738.76		4738.93		4738.6219		4738.577		**4738.575**
6477(85%)	1	0.7277	718.50	0.74	734.80	0.76089	752.485	0.73203	722.814	0.7279	718.683	0.7277	718.499
	2	0.6561	641.20	0.64	624.22	0.65494	640.005	0.6545	639.566	0.6556	640.729	0.6561	641.214
	3	1.0000	903.35	1.00	903.35	1.00000	903.345	0.9999	903.292	0.9999	903.344	0.9999	903.345
	4	1.0000	781.49	1.00	781.49	1.00000	781.489	0.9999	781.418	0.9999	781.486	0.9999	781.489
	5	1.0000	755.20	1.00	755.20	1.00000	755.201	0.9999	755.151	0.9999	755.200	0.9999	755.201
	6	0.7165	621.91	0.72	623.00	0.68377	590.051	0.71375	619.210	0.7168	622.204	0.7165	621.899
	∑		4421.65		4422.06		4423.03		4421.6687		4421.650		**4421.648**
6096(80%)	1	0.6591	653.5696	0.64	639.69	0.64049	637.192	0.64123	637.832	0.6420	638.562	0.6427	639.129
	2	0.5798	569.0161	0.55	545.77	0.54748	540.894	0.56088	552.370	0.5633	554.533	0.5626	553.915
	3	0.9991	902.8647	1.00	903.35	1.0000	903.345	0.9999	903.292	0.9999	903.344	0.9999	903.345
	4	0.9979	780.0799	0.998	780.27	0.99988	781.404	0.9999	781.418	0.9997	781.488	0.9999	781.489
	5	0.9921	751.2365	1.00	755.20	1.0000	755.201	0.9999	755.151	0.9997	755.200	0.9999	755.201
	6	0.5710	491.0385	0.61	519.83	0.61245	526.030	0.5979	513.529	0.5944	510.576	0.5941	510.626
	∑		4147.8055		4144.12		4144.06		4143.7452		4143.707		**4143.706**
5717(75%)	1	0.7713	763.4782	0.57	581.76	0.54808	563.853	0.55982	572.440	0.5583	571.321	0.5577	570.906
	2	0.7177	705.3382	0.46	468.94	0.41205	439.027	0.46897	478.710	0.4705	479.845	0.4691	478.848
	3	0.3000	292.0994	1.00	903.35	1.0000	903.345	0.9999	903.292	0.9999	903.344	0.9999	903.345
	4	0.9991	780.8389	1.00	781.49	1.0000	781.489	0.9999	781.418	0.9993	781.062	0.9999	781.489
	5	1.0000	755.2010	1.00	755.20	1.0000	755.201	0.9999	755.151	0.9999	755.200	0.9999	755.201
	6	0.7187	624.0084	0.47	415.46	0.54280	468.202	0.4724	414.919	0.4727	415.151	0.4724	414.959
	∑		3920.9642		3906.19		3911.118		3906.192		3905.947		**3904.748**
5334(70%)	1	0.6418	638.3097	0.63	632.28	0.74429	735.262	0.6686	662.111	0.6734	666.558	0.6726	582.047
	2	0.6621	647.2355	0.60	588.62	0.60361	590.645	0.6019	589.064	0.5943	582.095	0.5955	583.213
	3	0.3301	328.5020	0.30	292.10	0.3000	292.099	0.3000	292.099	0.3000	292.108	0.3000	292.099
	4	0.9906	774.8633	1.00	781.49	1.0000	781.489	0.9999	781.418	0.9999	781.468	0.9999	781.489
	5	0.9990	754.6915	1.00	755.20	1.0000	755.201	0.9999	755.151	0.9999	755.194	0.9999	755.201
	6	0.5806	498.9765	0.67	577.77	0.5557	478.529	0.6349	545.753	0.6378	548.359	0.6374	547.991
	∑		3642.5786		3627.46		3633.226		3625.814		3625.785		**3625.770**

Table 4. Statistical results generated in 100 consecutive algorithm runs

Heat load CL (kW)	Algorithm	Best	Mean	Worst	Standart dev.
6858 (90%)	ACS	4738.575301	4738.575400	4738.576158	1.7238E-4
	CS	4738.577912	4738.579036	4738.579916	7.0237E-4
	QPSO	4738.621934	4799.510137	4978.155780	55.205631
	ABC	4738.932446	4806.289173	4936.003941	39.616563
6477(85%)	ACS	4421.648633	4421.648683	4421.649083	8.4311E-5
	CS	4421.650780	4421.651565	4421.651912	3.7175E-4
	QPSO	4421.668748	4494.568310	4712.825474	59.586641
	ABC	4423.061152	4498.713943	4686.469043	36.729265
6096(80%)	ACS	4143.706369	4143.706540	4143.708741	3.7051E-4
	CS	4143.707973	4143.809456	4143.709995	6.3988E-4
	QPSO	4143.745256	4224.403932	4411.727210	60.011907
	ABC	4144.059228	4256.115154	4378.339028	44.621583
5717(75%)	ACS	3904.748508	3904.748611	3904.749245	1.7936E-4
	CS	3905.947164	3905.987621	3905.999005	0.0156959
	QPSO	3906.192063	3965.530031	4117.842967	40.765232
	ABC	3911.152708	3995.767210	4099.429807	37.655629
5334(70%)	ACS	3625.770345	3625.770373	3625.770712	6.7392E-5
	CS	3625.791938	3625.794996	3625.797901	0.0017250
	QPSO	3625.814945	3702.861754	3853.774782	46.541856
	ABC	3633.301740	3848.056379	3855.774086	33.695005

Table 5. CPU times (in seconds) for case study 1

	90%	85%	80%	75%	70%	Average calc. time
ACS	0.378	0.375	0.375	0.431	0.381	0.3880
ABC	0.131	0.132	0.134	0.133	0.134	0.1328
QPSO	0.330	0.328	0.327	0.326	0.331	0.3284
CS	51.232	53.762	57.843	49.592	43.785	51.2428

5.2 Case study 2: Four chillers

This case is subjected to a hotel which has two 450 RT and two 1000 RT units [2]. Optimizers those were previously utilized for this case are Genetic Algorithm (GA)[2], Lagrangian Method (LGM)[2], Particle Swarm Optimization (PSO) [8], and Differential Evolution (DE)[8]. Table 6 gives the best solutions of AVL, GA, LGM, ABC, QPSO, CS and ACS when power load shifts from 90% to 40% of total system load. DE and PSO are not considered in this table since infeasible constraints were assigned to these algorithms as surge may happen in these conditions. Table 6 reports that in 90% power load, input power for AVL is 2050.509 kW while that in ACS algorithm is 1857.298 kW which means 193.211 kW power is saved. Power savings for 80%, 70%, 60%, 50%, and 40% of total system loads are 74.316 kW, 14.223 kW, 3.558 kW, 26.024 kW and 71.362 kW respectively. These values are also plotted in Figure 3. Table 7 compares the statistical results for case study 2 including minimum value, maximum value, mean value and standard deviation. From the statistical comparisons, it is concluded that ACS shows its superiority on other algorithms in terms of solution quality and solution accuracy since ACS finds almost the same solution in each algorithm run with a smaller objective function value. Average execution speeds for the ACS, ABC, QPSO, and CS are 0.144, 0.1275, 0.2301, 29.508 seconds, as listed in Table 8.

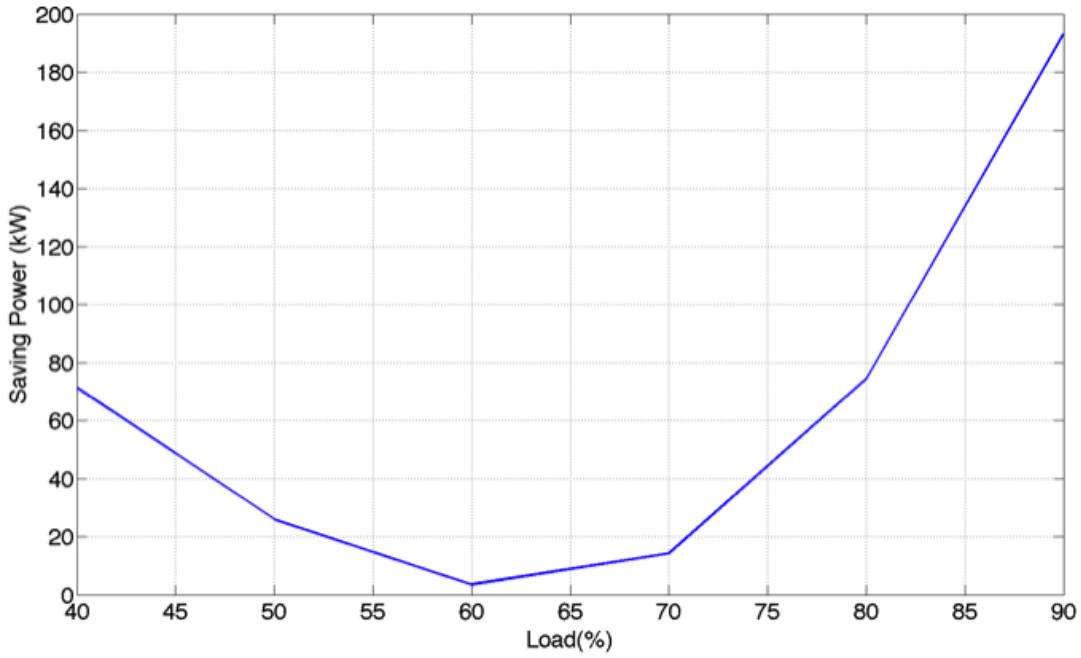


Figure 3. Power saving plot for case study 2

Table 6. Optimal chiller loading values in 100 runs for each algorithm

RT	Chiller	AVL		LGM [2]		GA [2]		ABC		QPSO		CS		ACS	
	i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i
2610(90%)	1	0.9000	279.23	0.9909	345.43	0.9925	346.77	0.9926	346.926	0.9908	345.433	0.9903	345.043	0.9908	345.433
	2	0.9000	293.30	0.9059	298.07	0.9487	336.73	0.9067	298.778	0.9058	298.071	0.9063	298.478	0.9058	298.071
	3	0.9000	570.07	1.0000	693.80	1.0000	693.80	1.0000	693.800	0.9999	693.800	1.0000	693.800	0.9999	693.800
	4	0.9000	907.90	0.7565	519.99	0.7366	485.68	0.7552	517.804	0.7564	519.993	0.7564	519.977	0.7564	519.993
	∑		2050.509		1857.30		1892.18		1857.311		1857.298		1857.299		**1857.298**
2320(80%)	1	0.8000	224.48	0.8289	238.52	0.8611	255.83	0.8337	241.023	0.8285	238.367	0.8287	238.463	0.8288	238.515
	2	0.8000	229.13	0.8056	231.92	0.8132	235.89	0.8158	237.316	0.8050	231.677	0.8059	232.129	0.8055	231.921
	3	0.8000	465.96	0.8966	566.19	0.8809	548.70	0.8912	560.181	0.8968	566.505	0.8964	566.034	0.8965	566.187
	4	0.8000	610.42	0.6879	419.04	0.6859	416.80	0.6864	417.346	0.6879	419.115	0.6879	566.034	0.6879	419.040
	∑		1529.976		1455.66		1457.23		1455.867		1455.665		1455.665		**1455.664**
2030(70%)	1	0.7000	185.72	0.7262	326.79	0.6592	173.81	0.7481	194.502	0.7248	194.027	0.7259	194.419	0.7262	194.502
	2	0.7000	191.37	0.7402	333.10	0.7605	211.55	0.7484	203.938	0.7400	203.873	0.7401	203.912	0.7402	203.938
	3	0.7000	381.84	0.7216	721.58	0.7557	426.22	0.7073	398.278	0.7217	398.381	0.7219	398.599	0.7215	398.278
	4	0.7000	433.42	0.6485	648.53	0.6360	372.23	0.6491	381.417	0.6490	381.857	0.6482	381.206	0.6485	381.417
	∑		1192.358		1178.14		1183.80		1178.137		1178.139		1178.137		**1178.137**
1740(60%)	1	0.6000	159.89	0.6035	271.60	0.5956	159.89	0.6100	161.986	0.5981	159.524	0.6042	160.770	0.6035	160.620
	2	0.6000	171.30	0.6576	295.93	0.6982	171.30	0.6629	182.334	0.6598	181.676	0.6583	181.364	0.6576	181.215
	3	0.6000	318.09	0.5648	564.79	0.5710	318.09	0.5673	301.772	0.5657	301.019	0.5643	300.366	0.5647	300.568
	4	0.6000	352.80	0.6077	607.68	0.5874	352.80	0.5997	352.716	0.6081	356.332	0.6074	356.032	0.6076	356.127
	∑		1002.089		998.53		1001.62		998.810		998.551		998.534		**998.532**
1450(50%)	1	0.5000	143.91	0.4590	139.49	0.5962	159.14	0.5251	145.289	0.5269	147.425	0.5250	147.155	0.5251	147.174
	2	0.5000	160.18	0.4985	160.03	0.3636	143.63	0.3000	129.805	0.3000	129.805	0.3000	129.805	0.3000	129.805
	3	0.5000	275.10	0.4471	260.88	0.4423	259.89	0.4922	275.519	0.4898	271.908	0.4924	272.709	0.4922	272.659
	4	0.5000	344.42	0.5721	344.23	0.5758	345.05	0.5863	347.148	0.5880	340.635	0.5862	347.916	0.5863	347.946
	∑		923.608		904.62		907.72		897.769		897.603		897.586		**897.586**
1160(40%)	1	0.4000	134.70	0.3000	129.19	0.3335	130.81	0.3000	129.187	0.3000	129.187	0.3000	129.187	0.3000	129.187
	2	0.4000	149.26	0.3000	129.81	0.3157	133.79	0.3000	129.812	0.3000	129.805	0.3000	129.805	0.3000	129.805
	3	0.4000	253.24	0.3514	250.36	0.3246	250.96	0.3514	250.368	0.3511	250.359	0.3513	250.360	0.3514	250.361
	4	0.4000	384.15	0.5386	340.63	0.5436	340.74	0.5385	340.630	0.5388	340.635	0.5386	340.634	0.5385	340.634
	∑		921.350		849.99		856.30		849.999		849.988		849.988		**849.988**

Table 7 Statistical results generated in 100 runs for case study 2.

Heat load CL (kW)	Algorithm	Best	Mean	Worst	Standart dev.
2610(90%)	ACS	1857.29863	1857.29863	1857.29863	2.27E-13
	QPSO	1857.29863	1858.44869	1880.03026	3.356326
	CS	1857.29963	1857.30059	1857.30098	4.112E-4
	ABC	1857.31156	1858.49879	1861.82285	0.945301
2320(80%)	ACS	1455.66474	1455.66474	1455.66474	1.10E-12
	CS	1455.66503	1455.66519	1455.66583	2.179E-4
	QPSO	1455.66522	1462.77787	1514.64243	10.27549
	ABC	1455.86767	1481.27779	1544.18159	15.64336
2030(70%)	ACS	1178.13701	1178.13701	1178.13701	3.89E-13
	CS	1178.13756	1178.13798	1178.13899	4.618E-4
	QPSO	1178.13963	1188.20687	1306.21515	15.79951
	ABC	1178.75776	1209.17160	1286.33111	21.01274
1740(60%)	ACS	998.53266	998.532662	998.532666	4.62E-13
	CS	998.53430	998.535703	998.535422	0.001547
	QPSO	998.55199	1007.13050	1045.99055	11.12384
	ABC	998.81027	1028.60505	1088.81295	20.03831
1450(50%)	ACS	897.58661	897.58661	897.58661	5.82E-13
	CS	897.58692	897.59174	897.59999	0.004626
	QPSO	897.60359	916.79371	1089.50611	27.90597
	ABC	897.76988	913.77723	952.50932	10.78515
1160(40%)	ACS	849.98823	849.98823	849.98823	2.84E-14
	CS	849.98827	849.98848	849.98899	2.684E-4
	QPSO	849.98842	883.66858	1055.35404	36.59615
	ABC	849.98843	862.14635	946.03202	13.71662

Table 8 CPU times (in seconds) for case study 2

	90%	80%	70%	60%	50%	40%	Average calc. time
ACS	0.145	0.143	0.142	0.148	0.142	0.144	0.1440
ABC	0.129	0.128	0.125	0.127	0.127	0.129	0.1275
QPSO	0.230	0.229	0.234	0.226	0.237	0.225	0.2301
CS	20.021	33.985	23.137	23.595	41.544	34.766	29.508

5.3 Case study 3: Three chillers

Third case study is concerned with a semiconductor plant in Hsin Tsu Science-based Park which has three 800 RT units. Firefly Algorithm [11], Improved Firefly Algorithm [11], Gradient Method (GM) [30], Genetic Algorithm (GA) [2], Particle Swarm Optimization [8] and Differential Evolution [9] are the optimization engines those were used for solving this case study. Table 9 demonstrates the results of optimal chiller loading acquired by AVL, GM, GA, ABC, QPSO, CS and ACS. The best results obtained by ACS are similar with those acquired by other algorithms as given in Table 9. Table 10 lists the statistical comparison results of ACS, CS, QPSO and ABC algorithms. As shown in Table 10, ACS is superior to other algorithms in terms of effectiveness and robustness. Power savings those occurred by using ACS algorithm substituted for average loading method (AVL) are plotted in Figure 4. From Figure 4, it is understood that ACS can save 2.264 – 34.008 kW of power as loading varies. Table 11 presents the average CPU times for the ACS, CS, ABC and QPSO algorithms. As seen in Table 11, CS shows the worst performance with an elapsed time of 39.9801 seconds.

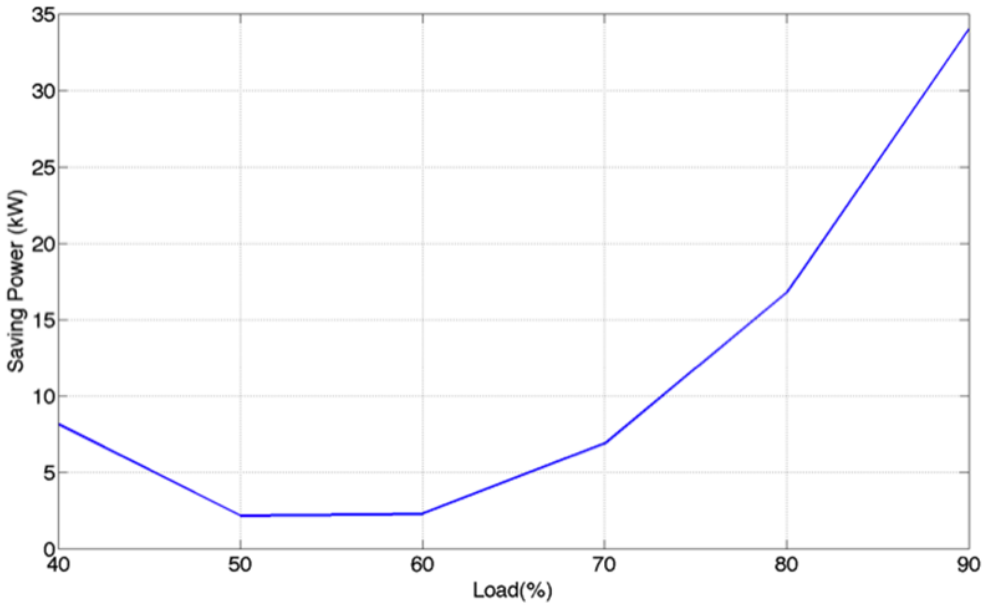


Figure 4. Power savings those occurred by using ACS algorithm substituted for average loading method (AVL)

Table 9. Results of the average loading and optimal chiller loading for case study 3

RT	Chiller	AVL		GM [30]		GA [2]		ABC		QPSO		CS		ACS	
	i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i	R_i	P_i
2160(90%)	1	0.9000	624.07	0.7253	483.45	0.8050	540.47	0.7253	483.477	0.7252	483.494	0.7253	483.533	0.7252	483.450
	2	0.9000	505.20	0.9747	551.59	0.9323	524.68	0.9746	551.497	0.9747	551.544	0.9746	551.508	0.9747	551.589
	3	0.9000	488.54	1.0000	548.77	0.9631	525.81	1.0000	548.767	0.9999	548.767	0.9999	548.747	0.9999	548.767
	∑		1617.814		1583.81		1590.96		1583.807		1583.806		1583.806		**1583.806**
1920(80%)	1	0.8000	536.58	0.6590	443.37	0.7017	468.49	0.6560	441.985	0.6590	443.368	0.6593	443.534	0.6590	443.368
	2	0.8000	449.39	0.8585	481.25	0.7954	446.97	0.8643	484.545	0.8584	481.235	0.8584	481.200	0.8585	481.248
	3	0.8000	433.99	0.8825	478.58	0.9035	490.56	0.8790	476.683	0.8824	478.591	0.8822	478.460	0.8824	478.578
	∑		1419.954		1403.20		1406.02		1403.217		1403.196		1403.196		**1403.196**
1680(70%)	1	0.7000	467.47	0.5962	410.09	0.6900	461.40	0.6065	415.325	0.5960	410.008	0.5955	409.760	0.5961	410.087
	2	0.7000	399.20	0.7450	421.18	0.6784	388.94	0.7467	422.045	0.7446	421.022	0.7444	420.934	0.7449	421.178
	3	0.7000	384.52	0.7589	413.06	0.7318	399.72	0.7467	407.037	0.7593	413.294	0.7600	413.630	0.7588	413.059
	∑		1251.187		1244.32		1250.06		1244.420		1244.325		1244.324		**1244.324**
1440(60%)	1	0.6000	412.01	0.5303	378.91	0.5217	375.05	0.5315	379.467	0.5307	379.102	0.5300	378.797	0.5303	378.913
	2	0.6000	359.95	0.6155	359.95	0.7407	419.04	0.6193	361.709	0.6159	360.160	0.6152	359.836	0.6165	359.949
	3	0.6000	363.40	0.6542	363.40	0.5381	313.66	0.6491	361.095	0.6533	363.001	0.6547	363.630	0.6542	363.401
	∑		1104.528		1102.26		1107.75		1102.274		1102.264		1102.264		**1102.264**
1200(50%)	1	0.5000	365.47	0.4986	364.86	0.4882	360.33	0.5035	367.003	0.5017	364.245	0.4974	364.369	0.4986	364.855
	2	0.5000	309.12	0.3849	259.34	0.4437	284.81	0.3778	256.237	0.3773	256.021	0.3892	261.226	0.3849	259.341
	3	0.5000	298.41	0.6164	346.55	0.5682	326.07	0.6186	347.611	0.6209	348.585	0.6132	345.255	0.6164	346.652
	∑		972.992		970.85		971.21		970.852		970.852		970.851		**970.849**
960(40%)	1	0.4000	323.11	0.3000	280.22	0.3055	282.66	0.3000	280.215	0.3000	280.215	0.3000	280.215	0.3000	280.215
	2	0.4000	265.90	0.3000	221.70	0.3185	230.02	0.3000	221.698	0.3000	221.698	0.3000	221.698	0.3000	221.698
	3	0.4000	260.58	0.6000	339.52	0.5764	329.51	0.6000	339.522	0.5999	339.522	0.5999	339.522	0.5999	339.522
	∑		849.591		841.44		842.18		841.436		841.436		841.436		**841.436**

Table 10 Statistical results obtained in 100 runs for case study 3

Heat load CL (kW)	Algorithm	Best	Mean	Worst	Standart dev.
2160(90%)	ACS	1583.80666	1583.80666	1583.80666	6.83E-13
	QPSO	1583.80667	1584.65548	1593.22580	1.671132
	CS	1583.80676	1583.80693	1583.80699	7.861E-5
	ABC	1583.80745	1586.78891	1644.38898	6.534132
1920(80%)	ACS	1403.19602	1403.19602	1403.19602	4.32E-13
	QPSO	1403.19602	1404.73123	1416.66572	2.19304
	CS	1403.19609	1403.19655	1403.19677	1.552E-4
	ABC	1403.21795	1408.08309	1438.46664	4.40645
1680(70%)	ACS	1244.32492	1244.32492	1244.32492	6.72E-13
	CS	1244.32498	1244.32773	1244.32898	7.930E-4
	QPSO	1244.32501	1246.05851	1262.28965	2.65294
	ABC	1244.42033	1248.46630	1274.28014	4.53585
1440(60%)	ACS	1102.26462	1102.26462	1102.26462	2.27E-13
	CS	1102.26482	1102.26623	1102.26699	4.752E-4
	QPSO	1102.26487	1104.14268	1117.02548	2.41906
	ABC	1102.27403	1105.02119	1122.93667	2.67466
1200(50%)	ACS	970.849932	970.849932	970.84993	5.710E-13
	CS	970.851165	970.854102	970.85497	0.0012473
	QPSO	970.852244	973.102904	988.77110	2.9025944
	ABC	970.855810	971.805533	980.08273	1.0093132
960(40%)	ACS	841.436119	841.436119	841.436119	2.273E-13
	CS	841.436119	841.436426	841.436499	8.1994E-5
	QPSO	841.436119	845.973666	856.615083	4.0053804
	ABC	841.436119	842.383515	850.364248	1.4481929

Table 11 CPU times (in seconds) for case study 3

	90%	80%	70%	60%	50%	40%	Average calc. time
ACS	0.121	0.123	0.121	0.134	0.124	0.123	0.1243
ABC	0.121	0.122	0.124	0.120	0.126	0.122	0.1225
QPSO	0.176	0.176	0.174	0.181	0.175	0.178	0.1766
CS	36.252	44.452	37.777	33.721	45.691	41.987	39.9801

6. Conclusion

This paper proposes Artificial Cooperative Search algorithm for optimal loading of multi-chiller system. ACS is a parameter free and swarm based algorithm which is constructed on interaction between prey and predator individuals while they are migrating to find more fruitful areas. Partial loading ratio is considered as design variable while optimization objective is to minimize total energy consumption of the multi chiller system. Three case studies adopted from the literature are used for benchmarking of ACS. In addition, Artificial Bee Colony, Quantum behaved Particle Swarm Optimization and Cuckoo Search are also applied to these case studies and results are compared with literature approaches. Comparison results indicate that ACS outperforms other algorithms in terms of robustness and effectivity. ACS also overcomes the convergence problem of Lagrangian method that takes place at low power demands. ACS is promising optimization method for multi-dimensional problems and proves its efficiency on improving the calculation performance of optimal chiller loading problem. For further investigations, ACS will be applied on multi-objective economic-emission load dispatch problems.

References

[1] Chang, Y.C. (2004). A novel energy conservation method-optimal chiller loading, Electric Power Systems Research, Vol.69, pp. 221–226.

[2] Chang, Y.C., Lin, J.K., and Chuang, M.H. (2005). Optimal chiller loading by genetic algorithm for reducing energy consumption, Energy and Buildings, Vol. 37, pp.147-155**.**

[3] Chang, Y.C. (2005). Genetic algorithm based optimal chiller loading for energy conservation, Applied Thermal Engineering, Vol. 25, pp. 2800–2815.

[4] Chang, Y.C., Chen, W.H., Lee, C.Y. and Huang, C.N. (2006). Simulated annealing based optimal chiller loading for saving energy, Energy Conversion and Management, Vol.47, pp. 2044–2058.

[5] Chang, Y.C. (2006). An innovative approach for demand side management-optimal chiller loading by simulated annealing, Energy, Vol.31, pp.1883–1896.

[6] Ardakani, A.J., Ardakani, F.F. and Hosseinian, S.H. (2008). A novel approach for optimal chiller loading using particle swarm optimization. Energy and Buildings, Vol.40, pp.2177–2187.

[7] Chang, Y.C. and Chen, W.H. (2009). Optimal chilled water temperature calculation of multiple chiller systems using Hopfield neural network for saving energy, Energy, Vol.34, pp.448–456.

[8] Lee, W.S. and Lin, L.C. (2009). Optimal chiller loading by particle swarm algorithm for reducing energy consumption, Applied Thermal Engineering, Vol.29, pp. 1730 –1734.

[9] Lee, W.S., Chen, Y.T. and Kao, Y. (2011). Optimal chiller loading by differential evolution algorithm for reducing energy consumption, Energy and Buildings, Vol.43, pp. 599–604.

[10] Beghi, A., Cecchinato, L., Cosi, G. and Rampazzo, M. (2012). A PSO-based algorithm for optimal multiple chiller systems operation. Applied Thermal Engineering, Vol.32, pp. 31–40.

[11] Coelho, L.S. and Mariani, V.C. (2013). Improved firefly algorithm approach applied to chiller loading for energy conservation. Energy and Buildings, Vol.59, pp. 273–278.

[12] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

[13] Sun, J., Feng, B. and Xu, W. (2004). Particle swarm optimization with particles having quantum behavior, In: Proceedings of Congress on Evolutionary Computation, Portland (OR, USA), 325–331.

[14] Sun, J., Xu, W. and Feng, B. (2004). A global search strategy of quantum - behaved particle swarm optimization, In: Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 291 – 294.

[15] Yang, X.S. and Deb, S. (2009). Cuckoo search via Lévy flights, World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE Publications, 210–214.

[16] ASHRAE (2000). ASHRAE Handbook – HVAC Systems and Equipment. Chapter 38. Atlanta, GA.

[17] ASHRAE (1999), ASHRAE Handbook – HVAC Systems and Equipment, Chapter 40, Atlanta, GA

[18] Oftadeh, R., Mahjoob, M.J. and Shariatpanahi, M. (2010). A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search, Computers and Mathematics with Applications, Vol. 60, pp. 2087-2098.

[19] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Boston.

[20] Storn, R. and Price, K. (1997). Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization,Vol.11, pp. 341–359.

[21] Colorni, A., Dorigo, M. and Maniezzo, V. (1991). Distributed optimization by ant colonies, Actes de la Premiere Conference Europeenne Sur la vie Artificiellei Paris, France, Elsevier Publishing, 134-142

[22] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983). Optimization by simulated annealing, Science, Vol. 220, pp. 671–680.

[23] Eberhart, R.C. and Kennedy, J. (1995). A new optimizer using particle swarm theory. in: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, 39–43.

[24] Geem, Z.W., Kim, J.H. and Loganathan, G.V. (2001). A new heuristic optimization algorithm: harmony search, Simulation, Vol.76, pp.60–68.

[25] Yang, X.S. (2009). Firefly algorithms for multimodal optimization, Stochastic Algorithms: Foundations and Applications, SAGA 2009. Lecture Notes in Computer Sciences, 169–178.

[26] Yang, X.S. (2010). A new metaheuristic bat-inspired algorithm. in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, 284, 65-74.

[27] Civicioglu, P. (2013). Artificial cooperative search algorithm for numerical optimization problems. Information Sciences, Vol.229, pp.58–76

[28] Geem, Z.W. (2011). Solution quality improvement in chiller loading optimization, Applied Thermal Engineering, Vol.31, pp.1848-1851

[29] Chang, Y.C., Lee, C.Y., Chen, C.R., Chou, C.J., Chen, W.H. and Chen, W.H. (2009). Evolution strategy based optimal chiller loading for saving energy, Energy Conversion and Management, Vol.50, pp. 132–139.

[30] Chang, Y.C., Chan, T.S. and Lee, W.S. (2010). Economic dispatch of chiller plant by gradient method for saving energy, Applied Energy, Vol.87, pp.1096–1101.