Ain Shams Engineering Journal (2014) xxx, xxx–xxx

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

MECHANICAL ENGINEERING

Design and economic investigation of shell and tube heat exchangers using Improved Intelligent Tuned Harmony Search algorithm

Oguz Emrah Turgut [a,*], Mert Sinan Turgut [b], Mustafa Turhan Coban [a]

[a] Department of Mechanical Engineering, Faculty of Engineering, Ege University, Bornova, Izmir, Turkey
[b] Department of Mechanical Engineering, Faculty of Engineering, Dokuz Eylul University, Tınaztepe, Izmir, Turkey

Received 8 November 2013; revised 1 April 2014; accepted 18 May 2014

KEYWORDS

Intelligent tuned harmony search;
Metaheuristic;
Shell and tube heat exchangers;
Thermal design

Abstract This study explores the thermal design of shell and tube heat exchangers by using Improved Intelligent Tuned Harmony Search (I-ITHS) algorithm. Intelligent Tuned Harmony Search (ITHS) is an upgraded version of harmony search algorithm which has an advantage of deciding intensification and diversification processes by applying proper pitch adjusting strategy. In this study, we aim to improve the search capacity of ITHS algorithm by utilizing chaotic sequences instead of uniformly distributed random numbers and applying alternative search strategies inspired by Artificial Bee Colony algorithm and Opposition Based Learning on promising areas (best solutions). Design variables including baffle spacing, shell diameter, tube outer diameter and number of tube passes are used to minimize total cost of heat exchanger that incorporates capital investment and the sum of discounted annual energy expenditures related to pumping and heat exchanger area. Results show that I-ITHS can be utilized in optimizing shell and tube heat exchangers.

1. Introduction

Heat exchangers are the manufactures used for transferring heat to one section to another. Shell and tube heat exchangers are the most common type which is almost utilized in every part of energy applications. Chemical and process industries, power generation, air conditioning and medical applications can be an example of their utilization [1,4]. As application of the other type of heat exchanger is increasing, shell and tube heat exchangers continue its popularity because of its versatility [2]. In shell and tube heat exchangers while one fluid flows across the tube banks, other runs through the tubes. Heat transfer takes place between the shell and the tube side fluids.

* Corresponding author. Address: Department of Mechanical Engineering, Ege University, Ege University Campus, 35100 Bornova, Izmir, Turkey. Tel.: +90 537299583.
E-mail addresses: oeturgut@hotmail.com (O.E. Turgut), sinanturgut@me.com (M.S. Turgut), turhan_coban@yahoo.com (M.T. Coban).

Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

2090-4479 © 2014 Production and hosting by Elsevier B.V. on behalf of Ain Shams University.
http://dx.doi.org/10.1016/j.asej.2014.05.007

Shell and tube heat exchangers consist of tubes, baffles, shell, front head, rear head, tube sheets and nozzles. Fig. 1 describes the main components of shell and tube heat exchanger. They can operate at high temperatures and pressures. They are easy to assemble as they need maintenance and cleaning [2]. Design of simple shell and tube heat exchanger (STHE) is a very tedious process as there are plenty of design variables that should meet for a given heat duty requirement and set of design constraints. Design process starts with deciding reference geometric configuration of a heat exchanger and assigning an allowable pressure drop for hot and cold medium. After that, design variables are defined with respect to design specification and physical properties of STHE so as to obtain a reliable heat transfer coefficient which helps to acquire satisfactory heat transfer surface. The choice of suitable design variable is a time consuming and tiresome process since there may be many trial and error solutions to meet the design requirements. To overcome this challenging issue, many researchers developed efficient methods.

Fettaka et al. [5] performed non dominated sorting genetic algorithm (NSGA-II) for optimizing shell and tube heat exchangers. Nine decision variables were considered as tube layout pattern, number of tube passes, baffle spacing, baffle cut, tube-to-baffle diameter clearance, shell-to-baffle diametrical clearance, tube length, tube outer diameter, and tube wall thickness. Multiple Pareto optimal solutions which were trade-off between the two objectives are introduced for designers. NSGA-II exposed better performance than the literature studies. Cost minimization of shell and tube heat exchanger is maintained by imperialist competitive algorithm (ICA) [6] by Hadidi et al. [7]. Obtained results showed that ICA can be applicable to shell and tube heat exchanger design problems. Chaudhuri et al. [8] used simulated annealing algorithm to optimize heat exchanger area to have a better heat transfer characteristics. Mizutani et al. [9] used mixed integer programming approach to optimize STHE as there are many discrete design variables like number of tubes and shell passes. Caputo et al. [10] used genetic algorithm to investigate the economic design of the shell and tube heat exchangers. And also there are many attempts to minimize the total cost of STHE's by genetic algorithm [3,11–16]. Differential Evolution (DE) strategies [17], Particle Swarm Optimization (PSO) [18], Artificial Bee Colony [19], Chaotic Quantum behaved Particle Swarm Optimization [20], and Biogeography-Based Optimization algorithm (BBO) [21] were also conducted to minimize total cost of shell and tube heat exchangers.

In this study, total cost of shell and tube heat exchanger is minimized by Improved Intelligent Tuned Harmony Search (I-ITHS) which is an upgraded version of Intelligent Tuned Harmony Search algorithm [22]. To the best of author's knowledge, ITHS has been never used in optimizing shell and tube heat exchangers. ITHS is an upgraded version of harmony search algorithm which is established on a musician who tries to find a pleasing harmony determined by an esthetic criterion. This study uses chaotic sequences generated by Henon map [23] substituted for Gaussian distributed random numbers and intensify on promising areas by local search strategies borrowed from Artificial Bee Colony [24] algorithm and Opposition Based Learning [25] method to upgrade the probing capability of ITHS algorithm. Literature survey exposes that designing shell and tube heat exchangers by using traditional approaches is not a convenient way to obtain satisfying results. Metaheuristic techniques can be an alternative approach of conventional methods as they are derivative free and they find global optimum by applying stochastic searches. I-ITHS technique is proposed for practitioners who aim to design shell and tube heat exchanger in an efficient way. The objectives of this study are (i) to introduce Intelligent Tuned Harmony Search algorithm, (ii) to upgrade ITHS algorithm by proposed methodologies (iii) to optimize heat exchanger in economic point of view, (iv) to compare the results of I-ITHS with other evolutionary algorithms. Rest of the paper is organized as follows, Section 2 gives the details of intelligent tuned harmony search algorithm, Section 3 describes the improvements made on ITHS algorithm, Section 4 details mathematical model and design formulations of shell and tube heat exchangers, Section 5 proposes the results of the current algorithm and comparison of findings with the other algorithms, and finally conclusion is maintained with Section 6.

2. Intelligent Tuned Harmony Search algorithm

2.1. Harmony search

Harmony search is a new emerged metaheuristic algorithm developed by Geem et al. [26]. It works with concept of seeking the best harmonies within possible pitches. As a musician is eager to find perfect state of harmony determined by aesthetic

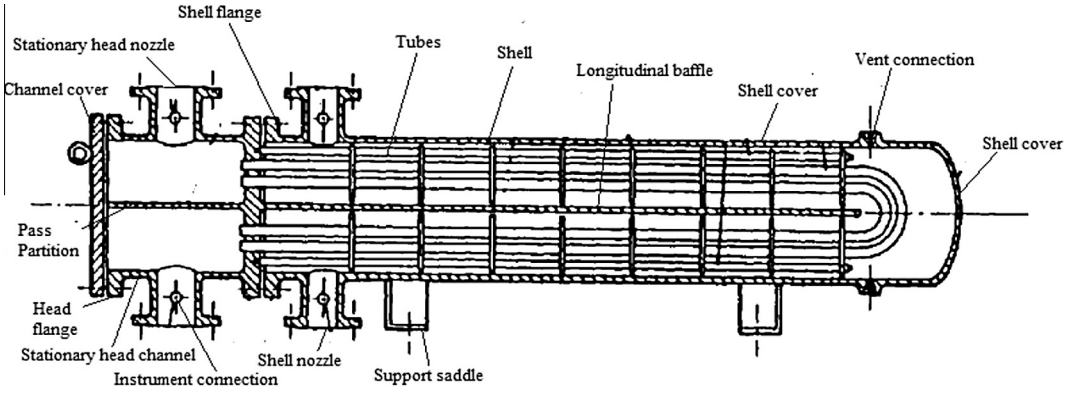

Figure 1 Representation of TEMA U-tube heat exchanger [2].

standard, objective function is in responsibility of deciding global best solution which is employed by optimization process. This algorithm is very useful does not need any derivative information, imposes fewer mathematical requirements and does not dependent of initial value estimation of design variables. Many engineering problems were solved by harmony search algorithm. Steel engineering problems [27,28], power and energy research [29–31], telecommunication [32–34], robotics [35], control systems [36], medical issues [37–39] and water system management [40] are the areas harmony search algorithm was exploited.

Harmony search algorithm can be conceptualized with describing the improvisation procedure of any musician. Improvisation of a new harmony can be sustained by three possible choices: a musician can play from his of her memory, play similar tune from his or her memory or uses random notes to compose a new harmony. These basic concepts are adapted to optimization procedure and applied within the steps that are presented below.

2.1.1. Initialization of optimization problem and algorithm parameters

$$Min\ f(x) \text{ with subject to } x_i \in [x_{LB}, x_{UB}], \quad i = 1, 2, \ldots, N$$

where $f(x)$ is the cost (objective) function, x is vector which holds the design variables, x_{LB} and x_{UB} are the lower and upper bounds of search space, respectively, and N is the number of design variables. There are several algorithm parameters that are necessary for solving optimization problem. These are namely harmony memory size (HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR) and termination criterion. Harmony memory size specifies the number of solution vector in harmony memory. HMS is generally populated between 30 and 100 solution vectors [41].

2.1.2. Initialization of harmony memory

Harmony memory (HM) is initialized by randomly generated solution vectors and their corresponding fitness values defined as

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 & | f(x_{N+1}^1) \\ x_1^2 & x_2^2 & \cdots & x_N^2 & | f(x_{N+1}^2) \\ \vdots & \vdots & \ddots & \vdots & | \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} & | f(x_{N+1}^{HMS}) \end{bmatrix} \tag{1}$$

2.1.3. Improvisation of a new harmony from the harmony memory (HM)

As it is said before, a new solution vector is generated in three possible ways which are based on harmony memory. These are memory consideration, pitch adjustment and randomization. Harmony memory consideration rate (HMCR) is concerned with memory consideration. With the possibility of HMCR, a solution vector is selected from HM. Otherwise, they are obtained randomly as the probability of 1-HMCR. These selections are represented as follows,

$$x_i \leftarrow \begin{cases} x_i \in \{x_i^1, x_i^2, \ldots, x_i^{HMS}\} \\ x_i \in [x_{LB}, x_{UB}] \end{cases} \tag{2}$$

PAR is the parameter which decides the rate of pitch adjustment. PAR is taken into practice if harmony memory consideration is utilized. Pitch adjustment strategy is organized as

$$x_i \leftarrow \begin{cases} x_i = x_i \pm rand()^* bw \\ x_i = x_i \end{cases} \tag{3}$$

where *rand*() is a uniformly distributed random number between 0 and 1, *bw* is an arbitrary distance bandwidth. For HMCR, parameter values should be varied between 0.7 and 0.95, and also for PAR, the values can be ranged between 0.2 and 0.5 so as to produce successful results [42].

2.1.4. Updating harmony memory

The new solution vector (harmony) is improvised as it is described in Step 3. Then harmonies are compared within the manner of their respective objective function. If the new improvised harmony is better than the worst one in HM, the worst is replaced with the new improvised harmony.

2.1.5. Checking the stopping criteria

Iterations are terminated if the maximum number of iteration is reached. Otherwise, Steps 3 to 5 will be repeated until preset conditions are satisfied.

2.2. Intelligent tuned harmony search algorithm

Yadav et al. [22] proposed a new variant of harmony search which is inspired by the decision making theory. In this algorithm, diversification and intensification phases are controlled by previous experience of the stochastic searches. Despotism rule is applied which dominates the algorithm to approach global best solution. In terms of despotism, harmony memory is divided into two groups. Groups are organized by the leader decided by the objective function value of the solution vector and can be represented as

$$\overrightarrow{x^{leader}} = HM(best, 1:N) \tag{4}$$

where *best* is the index of the best harmony in harmony memory. $\overrightarrow{x^{leader}}$, leader vector, takes the lead for deciding diversification and intensification properly. Leader vector is responsible for dividing the harmony memory into two groups as the procedure given in Table 1.

In this procedure, Group X is formed with solution vectors whose cost function is less than or equal to leader while Group Y includes the rest. HM^{mean} is the mean objective function value of the HM. Group X is in charge with intensification and diversification since Group Y is responsible with only diversification. Pitch adjustment rate (PAR) parameter is updated as proposed in Self Adaptive Harmony Search (SAHS) algorithm [43]

Table 1 Representation of group formation [22].

$HM^{mean} = mean(HM(:, N+1))$
for i = 1 to HMS
  if$(HM(i, N+1)) \le mean(HM(:, N+1))$
    HM$(i,:) \in$ *Group* X
  else
    HM$(i,:) \in$ *Group* Y
  end if
end for

Table 2 Pseudo code of proposed Improved Intelligent Tuned Harmony Search algorithm.

Initialize algorithm parameters:
Problem Dimension (D), Lower and Upper Bounds(L,U),
Pitch Adjustment Rate (PAR_{min}, PAR_{max})
Harmony Memory Size (HMS),
Harmony Memory Consideration Rate (HMCR)
Maximum Number of Iteration (Maxiter)
Set iteration counter (iter) to 0

Initialize Harmony Memory;
Calculate fitness value (fitval) of each harmony in Harmony Memory;
Find the best (best) and the worst (worst) harmonies in Harmony Memory (HM);

```
//Improvise Harmony Memory (HM)
1     while (iter < Maxiter)
2       for i = 1 to D do
3         if rand(0,1) < HMCR then // Memory consideration //
4           x_i = HM(d,i) where d ∈ [1,HMS]
5             PAR_iter = PAR_max − (PAR_max − PAR_min)(iter/Maxiter)
6               if rand(0,1) ⩽ PAR_iter then // Pitch adjustment
7     fitness_mean = mean(fitness);
8                 if (fitness_d < fitness_mean) then
9                 // Group X: Diversification and Intensification
10    if (rand(0,1) ⩽ 0.5) then
11                  y_i = best_i − (best_i − x_i) ∗ φ(0,1)
12                  else
13                  y_i = best_i + (worst_i − x_i) ∗ φ(0,1)
14                  endif
15                else
16                  //Group Y: Diversification
17                  best_d = L_i + (U_i − L_i)(best_d − L_d)/(U_d − L_d)
18                    y_i = x_i + (best_d − x_i) * φ(0,1)
19                endif
20                  if (L_i ⩽ y_i ⩽ U_i) then
21                  x_i = y_i
22                end
23              else
24              // random selection
25              x_i = L_i + (U_i − L_i) * φ(0,1)
26              end if
27            end for
            // Enhancing search procedure
            //Get the best harmony and apply local search
28            for j = 1 to D do
29              t_j = best_j + φ(0,1) * (best_j − HM(kk,i)) where kk ∈ [1,HMS] Λ kk ≠ HM_best
30            end
31              if (f(t) ⩽ f (best)) then
32              for j = 1 to D do
33                  best_j = t_j
34              end
35            end
          // opposition based learning
36          for j = 1 to D do
37            tt_j = U_j + L_j − best_j
38          end
39            if (f(tt) ⩽ f(best)) then
40            for j = 1 to D do
41              best_j = tt_j
42          end
43        end
44        Record the best harmony (best)
45        iter + +
46      end while
```

Please cite this article in press as: Turgut OE et al., Design and economic investigation of shell and tube heat exchangers using Improved Intelligent Tuned Harmony Search algorithm, Ain Shams Eng J (2014), http://dx.doi.org/10.1016/j.asej.2014.05.007

$$\mathrm{PAR(iter)} = \mathrm{PAR_{max}} - (\mathrm{PAR_{max}} - \mathrm{PAR_{min}})\left(\frac{iter}{\mathrm{Maxiter}}\right) \quad (5)$$

where $\mathrm{PAR_{max}}$ and $\mathrm{PAR_{min}}$ are maximum and minimum pitch adjustment rate values which are fixed to 1.0 and 0.0, respectively. Iteration and maxiter are iteration and maximum iteration values. This proposal extends the search capability of the algorithm and prevents the algorithm being trapped in local minimas. The pitch adjustment for selected design variable is maintained as follows

$$x_i \leftarrow \begin{cases} x_i^{best} - (x_i^{best} - x_i)rand() & \text{with probability of 0.5 PAR} \\ x_i^{best} + (x_i^{worst} - x_i)rand() & \text{with probability of 0.5 PAR} \\ x_i & \text{with probability 1-PAR} \end{cases} \quad (6)$$

where x_i^{best} and x_i^{worst} are ith design variable of the best and worst solution vectors. In (6) $x_i^{best} - (x_i^{best} - x_i)rand()$ term is responsible for intensification as x_i^{best} dominates the search concept and tends to intensify solution vector between the best and selected x_i. Another term in (6), $x_i^{best} + (x_i^{worst} - x_i)rand()$, is associated with diversification as x_i^{worst} controls the probability of selecting the design variable close or far away from x_i^{best}. All these pitch adjustment strategies belong to Group X that aims to achieve both diversification and intensification processes. As the search space of the Group X is limited between x^{best} and x^{worst}, algorithm may be trapped in local optimum. Group Y is generated to deal with this restriction. In Group Y, search capacity of the algorithm is enhanced with applying diversification. New x_i is selected between random elements of best solution vector $\overrightarrow{x^{best}}$ and previous x_i. Adapted pitch adjustment strategy is defined as

$$\begin{aligned} &x_i \leftarrow \{x_i + (x_m^{best} - x_i)rand() \\ &m \in [1, N] \end{aligned} \quad (7)$$

where m is randomly generated integer. Eq. (7) may lead to inefficient results as the search ranges of the design variables may change one dimension to other dimension. New design procedure is initiated to adapt x_m^{best} into search space of the upper and lower limits. Proposed $(x_m^{best})'$ is represented as

$$(x_m^{best})' = x_{L,i} + (x_{U,i} - x_{L,i})\frac{(x_m^{best} - x_{L,m})}{(x_{U,m} - x_{L,m})} \quad (8)$$

where $x_{L,i(m)}$ and $x_{U,i(m)}$ are the lower and upper bounds of the $i(m)$th design variable correspondingly.

3. Improved Intelligent Tuned Harmony Search algorithm

Harmony search algorithm is good at exploring overall search domain; however, it suffers from exploiting the promising areas obtained by diversification (exploration) process [44]. This deficiency is fixed with introducing Intelligent Tuned Harmony Search algorithm which balances the diversification and intensification of related optimization problem. Diversification refers to the ability of exploring unknown regions of search space while intensification uses priori knowledge to obtain better solutions. In this study, we propose a perturbed scheme adopted from Artificial Bee Colony (which is also utilized in [45]) to enhance global search capability of the algorithm. In addition, to ensure faster convergence and to upgrade solution quality, Opposition Based Learning method is employed.

3.1. Enhancing search mechanism with perturbed scheme

Inspired by Artificial Bee Colony algorithm, a new perturbed scheme is proposed as [45]

$$x_j = best_j + \phi_j(best_j - HM_{k,j}) \quad (9)$$

where $j = 1, \ldots, N$, *best* stands for the best harmony in harmony memory, ϕ is random number generated between -1.0 and 1.0 ($\phi \in [-1, 1]$) and $k \in [1, HMS]$ is a randomly generated integer which should be different from the best index of the harmony memory. This perturbation scheme takes the advantage of best index of the harmony memory and not only increases exploration capability of the algorithm but also helps to guide global best solution to another feasible solution in search domain. That is, it prevents from getting trapped in local minimum points. Nevertheless, as in line with "No free lunch theorem", convergence speed of the algorithm decreases due to the imposing search behavior of diversification process. This modification scheme is shown between 28 and 35 lines in Table 2.

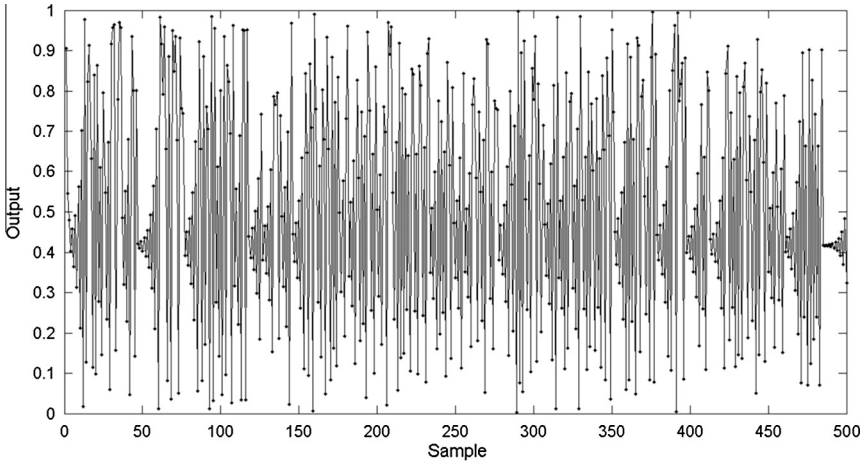

Figure 2 Evolution history of Henon map.

Table 3 Benchmark functions used to test the performance of Harmony Search based algorithms.

No	Range	Dimension (N)	Function	Formulation
f_1	$-5.12 \leqslant x_i \leqslant 5.12$	30–50	Sphere	$f(x) = \sum_{i=1}^{N} x_i^2$
f_2	$-2.048 \leqslant x_i \leqslant 2.048$	30–50	Rosenbrock	$f(x) = \sum_{i=1}^{N-1} \left[100(x_i^2 - x_{i+1}) + (x_i - 1)^2\right]$
f_3	$-5.12 \leqslant x_i \leqslant 5.12$	30–50	Rastrigin	$f(x) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]$
f_4	$-600.0 \leqslant x_i \leqslant 600.0$	30–50	Griewank	$f(x) = \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} \cos(x_i/\sqrt{i}) + 1$
f_5	$-10.0 \leqslant x_i \leqslant 10.0$	4	Colville	$f(x) = 100.0(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90.0(x_3^2 - x_4)^2 + 10.1((x_3 - 1)^2 + (x_4 - 1)^2) + 19.8(1.0/x_2)(x_4 - 1.0)$
f_6	$-10.0 \leqslant x_i \leqslant 10.0$	30–50	Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + e$
f_7	$-10.0 \leqslant x_i \leqslant 10.0$	30–50	Zakharov	$f(x) = \sum_{i=1}^{N} x_i^2 + \left(\sum_{i=1}^{N} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{N} 0.5 i x_i\right)^4$
f_8	$-100.0 \leqslant x_i \leqslant 100.0$	30–50	Penalized1	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k - (x_i - a)^m, & x_i < -a \end{cases} y_i = 1 + 0.25(x_i + 1)$ $f(x) = \frac{\pi}{N}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{N-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_N - 1)^2\right\} + \sum_{i=1}^{N} u(x_i, 10, 100, 4)$
f_9	$-600.0 \leqslant x_i \leqslant 600.0$	30–50	Step	$f(x) = \sum_{i=1}^{N} \lvert x_i + 0.5\rvert^2$
f_{10}	$-10.0 \leqslant x_i \leqslant 10.0$	30–50	Levy	$f(x) = \sin^2(\pi y_1) + \sum_{i=2}^{N-1}\left[\begin{array}{l}(y_i - 1)^2 \\ (1 + 10.0\sin^2(\pi y_i + 1.0))\end{array}\right] + (y_N - 1)^2(1.0 + 10.0\sin^2(2\pi y_N))$ $y_i = 1.0 + \frac{x_i - 1.0}{4}, i = 1, 2, \ldots, N$

3.2. Increasing the convergence speed of the algorithm by Opposition Based Learning

To conquer the drawback of slow convergence and to enhance the solution accuracy, Opposition Based Learning (OBL) method is employed here. This method is first proposed by Tizhoosh [25] and incorporated with many nature inspired algorithms such as Differential Evolution (DE) [51], Particle Swarm Optimization (PSO) [52], Ant Colony Optimization [53], Artificial Bee Colony Algorithm [54], etc. And also there are many varieties of OBL that include Quasi Opposition Based Learning (QOBL) [55] and Generalized Opposition-Based Learning (GOBL) [56], etc. More details about OBL can be found in [25,57]. OBL algorithm is based on the improving the current solution with its opposite. If opposite solution is better than the former one, it replaces with the former else, former ones remains same. OBL procedure is implemented as following:

In N – dimensional space, let $X = (x_1, x_2, \ldots, x_N)$ be a vector where $(x_1, x_2, \ldots, x_N) \in R$ and $x_i \in [L_i, U_i]$. The opposite point of x is defined as $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N)$ and each component of $\tilde{x}$ calculated with

$$\tilde{x}_i = L_i + U_i - x_i \tag{10}$$

where $i = 1, \ldots, N$; L and U represent the lower and upper bounds of the search space; x is former vector component. Suppose $f()$ is an objective function which measures the quality of the individuals. If $f(\tilde{x}) \leq f(x)$ x substitutes for $\tilde{x}$, else x remains unchanged. Opposition based search takes place between 36 and 43 lines in Table 2.

3.3. Increasing the diversity with Henon map [23]

Chaos is deterministic and random like process exists in nonlinear and dynamical systems, depends on the initial conditions and includes infinite motions [58]. Due to the non-repetition of the chaos, search speed of the chaotic algorithms is generally faster than stochastic ones. In literature, there is plenty of chaotic equations such as Tent map [59], Logistic map [60], Gauss map [61] which were also implemented in optimization algorithms [62–64] to increase the convergence ability and to enhance global search mechanism of related algorithm. In this article, we propose to utilize chaotic behavior of Henon map [23] with its ergodicity, irregularity and stochastic property instead of uniformly generated (Gaussian) random numbers. The use of chaotic sequences in replacement with uniformly distributed numbers may be feasible since it helps algorithm to escape from local minimum points and to increase the convergence rate of the algorithm. Henon map [23] proposed here is the simplified version of the Poincare map of Lorenz system and can be described as

$$\begin{aligned} y(t) &= 1 - a \cdot y(t-1) + z(t-1) \\ z(t) &= b \cdot y(t-1) \end{aligned} \tag{11}$$

For $a = 1.4$ and $b = 0.3$, Henon map is chaotic. In this condition, output values of $z(t) \in [-0.3854, 0.3819]$. Since we need numbers generated between 0 and 1, obtained output values are normalized in the range of $[0, 1]$. Fig 2 gives the normalized evolution history of Henon map. Generated chaotic numbers are symbolized as $\varphi(0, 1)$ in pseudo code of the proposed algorithm in Table 2.

3.4. Benchmarking of the proposed algorithm

To test the effectiveness of the proposed algorithm (I-ITHS), numerical experiments have been made with ten benchmark functions such as Sphere, Rosenbrock, Rastrigin, Griewank, Colville, Ackley, Zakharov, Penalized1, Step and Levy functions and statistical results of I-ITHS have been compared with Harmony Search [26] and its variants including Improved

Table 4 Statistical results of Harmony Search based algorithms generated in 100 runs (N = 30 D).

Functions	HS	IHS	GHS	NGHS	IGHS	MS	ITHS	I-ITHS
f_1 – Sphere								
Max	12.843	14.863	0.1209	7.12E−6	0.1839	6.42E−6	8.609E−4	0.00000
Std.dev	3.129	5.752	0.0672	7.45E−7	0.0732	2.13E−7	1.0788E−4	0.00000
Mean	6.178	7.183	0.0453	4.62E−7	0.1495	4.13E−8	1.5435E−5	0.00000
Min	0.976	1.263	3.23E−6	5.12E−8	9.27E−5	1.2E−34	9.5538E−32	0.00000
f_2 – Rosenbrock								
Max	7675.7852	3697.8871	3637.4565	5826.838	3724.9692	191.668	30.9743	28.5882
Std.dev	1324.0712	399.03013	379.0225	808.4382	380.5070	37.2354	0.00032	6.95872
Mean	6568.2721	2971.6793	2913.4565	2662.354	2974.591	64.9656	28.7276	21.2278
Min	3532.4677	1757.8977	1670.2319	1033.893	1838.033	15.6916	0.00108	0.00000
f_3 – Rastrigin								
Max	14.8039	16.9762	2.112E−1	100.931	3.312E−1	19.61E−10	15.5514	0.00000
Std.dev	2.875	3.173	3.823E−2	32.71213	6.971E−2	3.830E−11	3.206205	0.00000
Mean	6.726	7.862	2.87E−2	12.992	4.122E−2	6.766E−11	1.432211	0.00000
Min	2.1546	3.6722	7.726E−7	9.192E−9	8.127E−6	1.059E−12	5.68E−14	0.00000
f_4 – Griewank								
Max	1.21245	1.11225	3.22921	3.127E−1	1.11086	8.312E−12	9.908E−5	0.00000
Std.dev	0.02132	0.00772	1.922E−1	1.291E−2	0.18921	3.413E−14	1.706E−5	0.00000
Mean	1.18102	1.09762	2.098E−1	3.397E−2	0.10239	7.823E−16	4.684E−5	0.00000
Min	1.13398	1.07712	1.083E−6	1.253E−8	8.912E−6	4.123E−23	0.000000	0.00000
f_5 – Colville								
Max	1199.12	64.1334	70.0899	100.2162	56.7055	2.12432	46.6504	1.72412
Std.dev	176.813	11.3350	12.5434	32.7462	11.0777	1.22451	6.84032	0.33832
Mean	86.5421	21.7030	22.0271	41.3123	21.3872	0.03902	3.79251	0.18642
Min	2.05847	0.32821	0.67718	0.17176	0.59566	3.88E−7	5.58E−4	7.82E−17
f_6 – Ackley								
Max	1.83193	2.82121	1.32E−1	19.5456	28.0303	8.426E−9	0.002400	0.00000
Std.dev	9.722E−1	0.39702	1.92E−2	1.18772	2.43107	2.45E−10	4.6762E−4	0.00000
Mean	8.234E−1	1.2731	2.03E−2	7.8792	8.3734	3.77E−10	1.5512E−4	0.00000
Min	2.91E−2	6.28E−2	7.32E−6	8.734E−8	6.8641E−6	1.10E−12	7.993E−15	0.00000
f_7 – Zakharov								
Max	13.812	21.004	2.37912	0.8123	7.191E−2	1.79E−8	0.03610	0.00722
Std.dev	1.9321	3.0386	3.02E−3	9.02E−3	1.812E−4	3.42E−9	0.00537	0.00131
Mean	12.838	11.323	1.42E−2	2.68E−2	1.261E−3	8.21E−9	0.00189	3.41E−4
Min	11.323	1.63E−2	1.94E−5	7.31E−6	7.512E−6	1.42E−9	1.84E−11	6.3E−15
f_8 – Penalized1								
Max	2.69173	2.09411	3.932E−1	3.69E−2	3.664E−2	5.342E−5	5.4243E−4	3.08E−10
Std.dev	0.94291	0.338E−1	3.321E−3	6.18E−3	3.973E−3	1.434E−6	1.1507E−4	9.26E−10
Mean	1.27636	3.082E−3	3.143E−4	2.61E−4	3.027E−4	1.285E−6	2.8472E−5	6.61E−10
Min	0.51223	2.041E−3	2.490E−4	1.36E−4	1.957E−4	1.012E−8	3.107E−10	1.35E−10
f_9 – Step								
Max	2.26.6831	1.14754	1.13733	1.96434	1.32921	6.09E−10	1.153E−4	4.355E−5
Std.dev	0.207771	8.972E−2	7.93E−2	5.98E−2	0.88619	9.39E−11	2.576E−5	9.072E−6
Mean	1.831691	0.997544	0.99673	8.46E−2	5.792E−2	8.28E−11	9.259E−5	1.979E−7
Min	1.324612	0.768887	0.72113	2.12E−2	5.242E−2	4.81E−21	0.000000	0.00000
f_{10} – Levy								
Max	2.793E−1	1.223E−1	1.212E−2	1.492E−2	1.311E−2	2.312E−4	5.001E−5	7.076E−9
Std.dev	4.134E−2	1.128E−2	1.313E−3	2.685E−3	1.183E−3	2.215E−5	1.178E−5	9.04E−10
Mean	2.163E−1	1.043E−1	1.062E−2	9.641E−2	1.043E−2	2.673E−6	3.622E−6	4.868E−9
Min	1.392E−1	7.631E−2	8.276E−3	4.619E−3	7.582E−3	4.124E−7	5.764E−9	3.866E−9

Harmony Search (IHS) [46], Global best Harmony Search(GHS) [47], Novel Global Harmony Search (NGHS) [48], Improved Global Harmony Search (IGHS) [49], Melody Search (MS) [50] algorithm's. Statistical results are calculated over 100 runs for each algorithm both 30 and 50 dimensional problems. For each algorithm, HMCR = 0.95, HMS = 20, maximum number of improvisation (Maxiter) = 20,000, and, if necessary, PAR_{nim} and PAR_{max} are respectively set to 0.1 and 0.9. Computer code is developed with JAVA in a computer Intel Core CPU @ 2.50 GHz and 6.0 GB RAM. Table 3 gives formulations, upper and lower bounds of the aforementioned benchmark functions. Table 4 compares the

Table 5 Statistical results of the benchmark functions for ITHS and I-ITHS algorithms ($N = 50$ D).

	Min.	Mean	Std. dev.	Max	Min.	Mean	Std. dev.	Max	Min.	Mean	Std. dev.	Max
	f_1 – Sphere				f_2 – Rosenbrock				f_3 – Rastrigin			
I-ITHS	0.00000	0.00000	0.00000	0.00000	0.00000	9.1283	12.743	38.0081	0.00000	0.00000	0.00000	0.00000
ITHS	8.21E−6	1.37E−4	1.66E−4	8.64E−4	37.9519	38.1279	0.0981	38.2543	6.19E−5	2.2668	5.3072	19.241
	f_4 – Griewank				f_6 – Ackley				f_7 – Zakharov			
I-ITHS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	2.13E−16	0.2904	1.72142	13.5533
ITHS	3.024E−8	5.795E−6	8.418E−6	3.075E−6	2.684E−4	0.01711	0.01883	0.06421	3.612E−8	7.406E−4	0.00379	0.00348
	f_8 – Penalized1				f_9 – Step				f_{10} – Levy			
I-ITHS	8.69E−11	3.565E−8	7.939E−8	3.124E−7	2.31E−31	2.206E−6	8.592E−6	4.347E−5	3.951E−9	1.063E−4	4.715E−4	0.00189
ITHS	4.239E−8	9.984E−7	1.083E−6	3.593E−6	3.396E−7	8.645E−5	9.735E−5	3.981E−4	3.456E−8	1.142E−4	1.173E−4	4.128E−4

statistical results considering all mentioned harmony search based algorithms. From Table 4, it is clearly seen that I-ITHS algorithm outperforms other algorithms in all cases. Table 5 gives the comparison of the statistical results of ITHS and I-ITHS algorithms for 50 Dimensional problems. I-ITHS shows its superiority over ITHS algorithm since it yields better results than ITHS algorithms. In addition, although 50 Dimensional problems are considered, I-ITHS algorithm has found the global optimum of Rosenbrock function which is very challenging and hard-to-solve. Fig 3 gives the convergence history of 30 Dimensional Ackley, Rastrigin and Penalized1 functions for both ITHS and I-ITHS algorithms. It is clearly observed that I-ITHS algorithm has converged the optimum much faster than ITHS algorithm for all cases.

4. Mathematical modeling of shell and tube heat exchangers

Surface area of the heat exchanger is calculated by [65,66]

$$S = \frac{Q}{U\Delta T_{LM}F} \tag{12}$$

where Q is heat load, U is overall heat transfer coefficient, ΔT_{LM} is the logarithmic mean temperature difference and F is its correction factor. Then sensible heat transfer is acquired by

$$Q = m_s C_{ps}(T_{is} - T_{os}) = m_t C_{pt}(T_{ot} - T_{it}) \tag{13}$$

Overall heat transfer coefficient is the function of shell and tube side heat transfer coefficient and fouling resistance [10].

$$U = \frac{1}{\frac{1}{h_s} + R_{fs} + \frac{d_o}{d_i}\left(R_{ft} + \frac{1}{h_t}\right)} \tag{14}$$

$$d_i = 0.8d_o \tag{15}$$

Heat transfer coefficient for shell side is implemented by Kern's formulation [65]

$$h_s = 0.36\frac{k_s}{D_e}\mathrm{Re}_s^{0.55}\mathrm{Pr}_s^{0.33}\left(\frac{\mu_t}{\mu_w}\right)^{0.14} \tag{16}$$

where D_e is hydraulic shell diameter calculated as for square and triangular pitches [65,66].

For square pitch

$$D_e = \frac{4\left(P_t^2 - (0.25\pi d_o^2)\right)}{\pi d_o} \tag{17}$$

For triangular pitch

$$D_e = \frac{0.4\left(0.43P_t^2 - (0.5\pi d_o^2/4)\right)}{\pi d_o} \tag{18}$$

Prandtl and Reynolds number for shell side is computed as follows

$$\mathrm{Pr}_s = \frac{\mu_s C_{ps}}{k_s} \tag{19}$$

$$\mathrm{Re}_s = \frac{\rho_s v_s D_e}{\mu_s} \tag{20}$$

where v_s is the velocity of the fluid for the shell side and calculated by [65,66]

$$v_s = \frac{m_s}{a_s\rho_s} \tag{21}$$

In (21), a_s is the cross-sectional area normal to the flow and computed by

$$a_s = \frac{D_s \cdot B \cdot C_l}{\rho_s} \tag{22}$$

where C_l is the shell side clearance and calculated by

$$C_l = P_t - d_o \tag{23}$$

Tube side heat transfer coefficient is computed by the function of flow regime and represented as [65,67]

$$h_t = \frac{k_t}{d_i}\left[3.657 + \frac{0.0677\left(\mathrm{Re}_t\mathrm{Pr}_t\frac{d_i}{L}\right)^{1.33}}{1 + 0.1\mathrm{Pr}_t\left(\mathrm{Re}_t\frac{d_i}{L}\right)^{0.3}}\right] \tag{24}$$

$(\mathrm{Re}_t < 2300)$

$$h_t = \frac{k_t}{d_i}\left[\frac{\frac{f_t}{8}(\mathrm{Re}_t - 1000)\mathrm{Pr}_t}{1 + 12.7\sqrt{\frac{f_t}{8}}\left(\mathrm{Pr}_t^{0.66} - 1\right)}\left[1 + \left(\frac{d_i}{L}\right)^{0.67}\right]\right] \tag{25}$$

$(2300 < \mathrm{Re}_t < 10{,}000)$

$$h_t = 0.027\frac{k_t}{d_i}\mathrm{Re}_t^{0.8}\mathrm{Pr}_t^{0.33}\left(\frac{\mu_t}{\mu_w}\right)^{0.14} \tag{26}$$

$(\mathrm{Re}_t > 10{,}000)$ and Darcy friction factor is calculated with [68]

$$f_t = (1.82\log_{10}\mathrm{Re}_t - 1.64)^{-2} \tag{27}$$

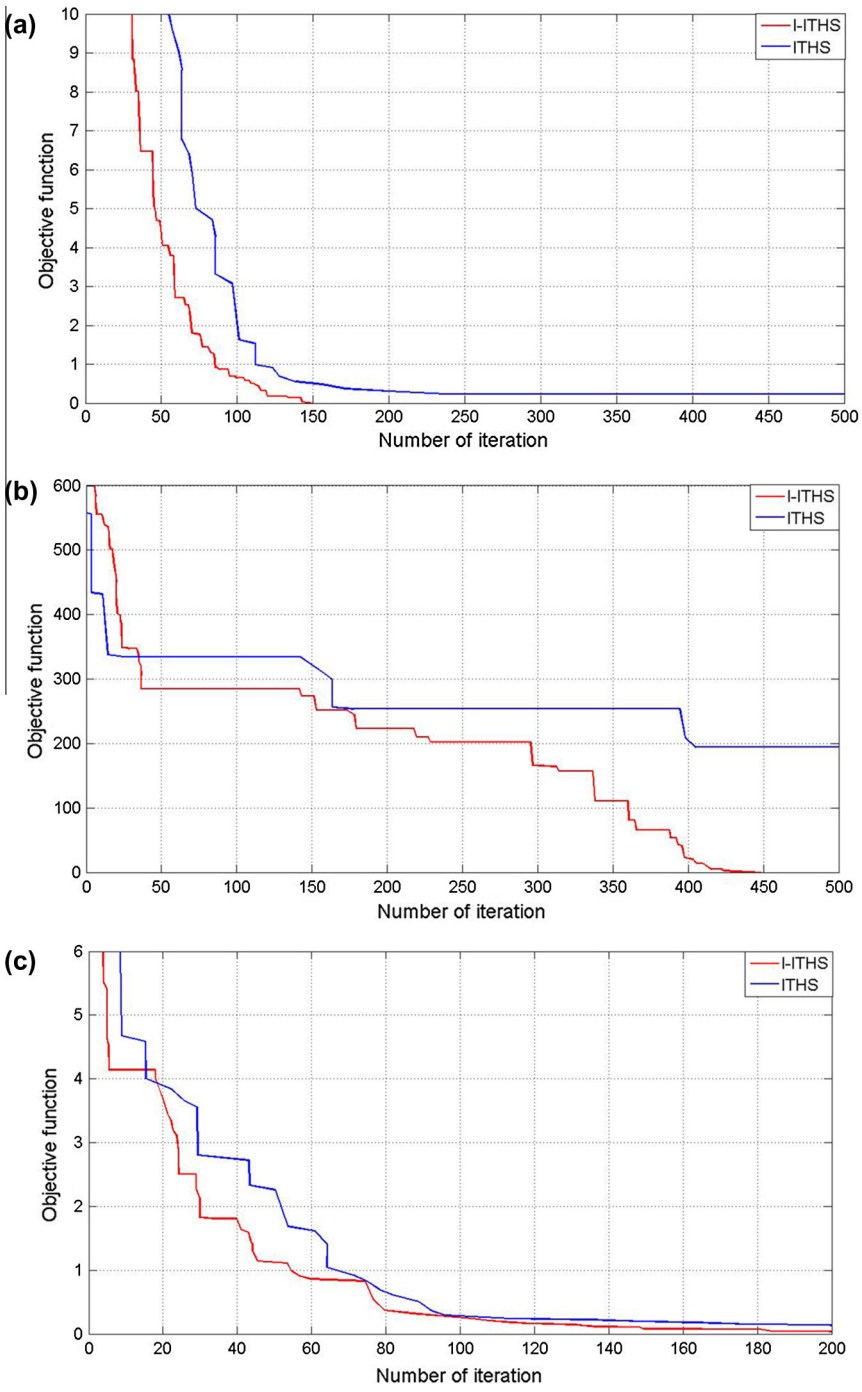

Figure 3 Comparison of the convergence speed of ITHS and I-ITHS algorithm by utilizing Ackley, Rastrigin and Penalized1 functions.

Reynolds and Prandtl numbers for the tube side is calculated by

$$\mathrm{Pr}_t = \frac{\mu_t C_{pt}}{k_t} \tag{28}$$

$$\mathrm{Re}_t = \frac{\rho_t v_t d_i}{\mu_t} \tag{29}$$

and tube side flow velocity is represented by [65]

$$v_t = \frac{m_t}{\frac{\pi d_i^2}{4}\rho_t}\frac{n}{N_t} \tag{30}$$

where n is the number of tube passes and N_t is the number of tubes that can be calculated by [66,69,70]

$$N_t = K_1\left(\frac{D_s}{d_o}\right)^{n_1} \tag{31}$$

K_1 and n_1 are the coefficients that are the functions of flow arrangement and number of passes. Table 6 shows the appropriate values for different flow arrangements.

Fig. 4 shows the triangular and square tube pitches. According to equations those are explained above, length of the tube is calculated with

Table 6 $K1$ and $n1$ coefficients for different pitch types [66].

Number of tube passes	Triangle tube pitch $P_t = 1.25\ d_o$		Square tube pitch $P_t = 1.25\ d_o$	
	C	n_1	C	n_1
1	0.319	2.142	0.215	2.207
2	0.249	2.207	0.156	2.291
4	0.175	2.285	0.158	2.263
6	0.0743	2.499	0.0402	2.617
8	0.0365	2.675	0.0331	2.643

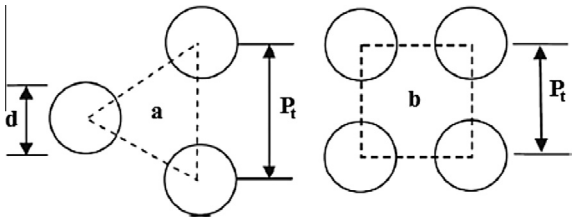

Figure 4 Representation of triangular and square tube pitch arrangement (a) triangle and (b) square.

$$L = \frac{S}{\pi d_o N_t} \tag{32}$$

Logarithmic mean temperature difference (ΔT_{LM}) which is described in (12) is computed by

$$\Delta T_{LM} = \frac{(T_{is} - T_{ot}) - (T_{os} - T_{it})}{\ln\left(\frac{(T_{is}-T_{ot})}{(T_{os}-T_{it})}\right)} \tag{33}$$

The correction factor F is used in (12) is calculated by the following procedure [71,72]

$$F = \frac{\sqrt{R^2+1}}{R-1}\frac{\ln\left(\frac{1-P}{1-PR}\right)}{\ln\left(\frac{2-P\left(R+1-\sqrt{R^2+1}\right)}{2-P\left(R+1+\sqrt{R^2+1}\right)}\right)} \tag{34}$$

where R and P correction and efficiency factors and dependent of flow configuration. For the configurations those are discussed in this article, they are given as

$$R = \frac{T_{is} - T_{os}}{T_{ot} - T_{it}} \tag{35}$$

$$P = \frac{T_{ot} - T_{it}}{T_{is} - T_{it}} \tag{36}$$

Pressure drop in heat exchanger is key design feature as there is a close relationship between heat transfer and pressure drop. For a constant heat duty in a specified heat exchanger as fluid velocity increases, heat transfer coefficient rises and this results the lower investment cost. However, increase of the fluid velocity causes more pressure drop which raises operational cost values. For that reason, the designer should select appropriate design variables adding to pressure drop restrictions.

The tube side pressure drop is composed of major and minor losses and presented as

$$\Delta P_t = \Delta P_{major} + \Delta P_{\min or} = \frac{\rho_t v_t^2}{2}\left(\frac{L}{d_i} + p\right)\cdot n \tag{37}$$

Different values are proposed for p constant by different researchers. Kern [65] considered $p = 4$, since Sinnot [66] assumed $p = 2.5$. The shell side pressure drop is calculated by [10,65]

$$\Delta P_s = f_s\left(\frac{\rho_s v_s^2}{2}\right)\left(\frac{L}{B}\right)\left(\frac{D_s}{D_e}\right) \tag{38}$$

and friction factor, f_s, is represented as follows

$$f_s = 2b_o \mathrm{Re}_s^{-0.15} \tag{39}$$

where $b_o = 0.72$ [72]. Pumping power is related with pumping efficiency and computed as

$$P = \frac{1}{\eta}\left(\frac{m_t}{\rho_t}\Delta P_t + \frac{m_s}{\rho_s}\Delta P_s\right) \tag{40}$$

Objective function for this minimization problem is total cost of heat exchanger (C_{tot}) which is the function of capital investment (C_i), energy cost (C_e), annual operating cost (C_o) and total discounted operating cost (C_{oD}) [10].

$$C_{tot} = C_i + C_{oD} \tag{41}$$

According to Hall's correlation [73], capital investment (C_i) is calculated by

$$C_i = a_1 + a_2 S^{a_3} \tag{42}$$

where $a_1 = 8000$, $a_2 = 259.2$ and $a_3 = 0.91$ are the coefficients for the heat exchangers made of stainless steel. The total discounted operating cost is established on to overcome friction losses and calculated by the following

$$C_o = P \cdot C_e \cdot H \tag{43}$$

$$C_{oD} = \sum_{k=1}^{ny} \frac{C_o}{(1+i)^k} \tag{44}$$

Table 7 Lower and upper bounds for design variables.

	Lower bound	Upper bound
Tube outside diameter (d_o) (m)	0.010	0.051
Shell side diameter (D_e) (m)	0.1	1.5
Baffle spacing (B) (m)	0.05	0.5
Number of tube passes	1	8

Please cite this article in press as: Turgut OE et al., Design and economic investigation of shell and tube heat exchangers using Improved Intelligent Tuned Harmony Search algorithm, Ain Shams Eng J (2014), http://dx.doi.org/10.1016/j.asej.2014.05.007

Table 8 Physical properties for different cases.

	m (kg/s)	T_i (°C)	T_o (°C)	ρ (kg/m^3)	C_p (kJ/kg K)	μ (Pa s)	k (W/mK)	R_f (m^2 K/W)
Case 1								
Shell side: methanol	27.8	95.0	40.0	750.0	2.84	0.00034	0.19	0.00033
Tube side: sea water	68.9	25.0	40.0	995.0	4.2	0.00080	0.59	0.00020
Case 2								
Shell side: kerosene	5.52	199	93.3	850.0	2.47	0.00040	0.13	0.00061
Tube side: crude oil	18.8	37.8	76.7	995.0	2.05	0.00358	0.13	0.00061
Case 3								
Shell side: distilled water	22.07	33.9	29.4	995	4.18	0.00080	0.62	0.00017
Tube side: raw water	35.31	23.9	26.7	999	4.18	0.00092	0.62	0.00017

Table 9 Comparison of the algorithms for case study 1.

	Original study [66]	GA [10]	PSO [18]	ABC [19]	BBO [21]	ITHS	I-ITHS
D_s (m)	0.894	0.830	0.81	1.3905	0.801	0.7620	0.7635
L (m)	4.830	3.379	3.115	3.963	2.040	2.0791	2.0391
B (m)	0.356	0.500	0.424	0.4669	0.500	0.4988	0.4955
d_o (m)	0.020	0.016	0.015	0.0104	0.010	0.0101	0.0100
P_t (m)	0.025	0.02	0.0187	–	0.0125	0.1264	0.0125
C_1 (m)	0.005	0.004	0.0037	–	0.0025	0.0253	0.0025
n	2.0	2.0	2.0	2.0	2.0	2.0	2.0
N_t	918.0	1567	1658	1528	3,587	3,454	3,558
v_t (m/s)	0.75	0.69	0.67	0.36	0.77	0.7820	0.7744
Re_t	14,925	10,936	10,503	–	7642.49	7842.52	7701.29
Pr_t	5.7	5.7	5.7	–	5.7	5.700	5.700
h_t (W/m^2 K)	3812	3762	3721	3818	4314	4415.918	4388.79
f_t	0.028	0.031	0.0311	–	0.034	0.03540	0.03555
ΔP_t (Pa)	6251	4298	4171	3043	6156	6998.70	6887.63
a_s (m^2)	0.0320	0.083	0.0687	–	0.0801	0.07602	0.07567
D_e (m)	0.014	0.011	0.0107	–	0.007	0.00719	0.00711
v_s (m/s)	0.58	0.44	0.53	0.118	0.46	0.48755	0.48979
Re_s	18,381	11,075	12,678	–	7254	7736.89	7684.054
Pr_s	5.1	5.1	5.1	–	5.1	5.08215	5.08215
h_s (W/m^2 K)	1573	1740	1950.8	3396	2197	2213.89	2230.913
f_s	0.330	0.357	0.349	–	0.379	0.3759	0.37621
ΔP_s (Pa)	35,789	13,267	20,551	8390	13,799	14,794.94	14,953.91
U (W/m^2 K)	615	660	713.9	832	755	760.594	761.578
S (m^2)	278.6	262.8	243.2	–	229.95	228.32	228.03
C_i (€)	51,507	49,259	46,453	44,559	44,536	44,301.66	44,259.01
C_o (€/year)	2111	947	1038.7	1014.5	984	964.164	962.4858
C_{oD} (€)	12,973	5818	6778.2	6233.8	6046	5924.373	5914.058
C_{tot} (€)	64,480	55,077	53,231	50,793	50,582	50,226	50,173

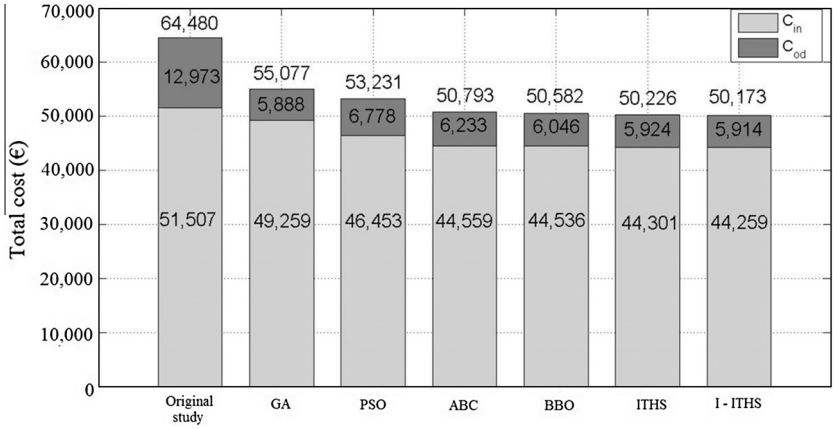

Figure 5 Cost comparison for case study 1.

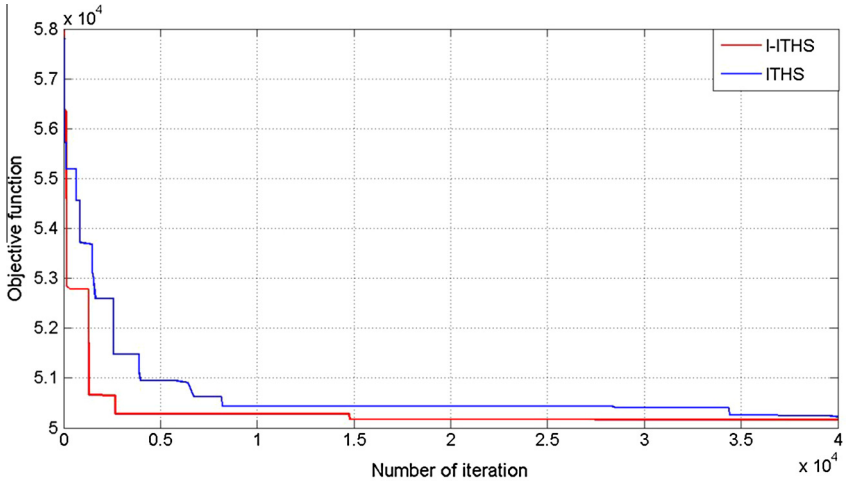

Figure 6 Convergence of ITHS for case study 1.

Table 10 Comparison of the algorithms for case study 2.

	Original study [65]	GA [10]	ITHS	I-ITHS
D_s (m)	0.539	0.63	0.32079	0.31619
L (m)	4.88	2.153	5.15184	5.06235
B (m)	0.127	0.12	0.24725	0.24147
d_o (m)	0.025	0.02	0.01204	0.01171
P_t (m)	0.031	0.025	0.01505	0.01464
C_1 (m)	0.006	0.005	0.00301	0.00293
n	4.0	4.0	1.0	1.0
N_t	158	391	301	309
v_t (m/s)	1.44	0.87	0.8615	0.8871
Re_t	8227	4068	2306.77	2303.46
Pr_t	55.2	55.2	56.4538	56.4538
h_t (W/m^2 K)	619	1168	1398.85	1435.68
f_t	0.033	1168	0.04848	0.04854
ΔP_t (Pa)	49,245	14,009	10,502.45	11,165.45
a_s (m^2)	0.0137	0.0148	0.01585	0.01527
D_e (m)	0.025	0.019	0.01188	0.01157
v_s (m/s)	0.47	0.43	0.40948	0.42526
Re_s	25,281	18,327	10,345.294	10,456.39
Pr_s	7.5	7.5	7.6	7.6
h_s (W/m^2 K)	920	1034	1248.86	1290.789
f_s	0.315	0.331	0.35987	0.35929
ΔP_s (Pa)	24,909	15,717	14,414.26	15,820.74
U (W/m^2 K)	317	376	326.071	331.358
S (m^2)	61.5	52.9	58.641	57.705
C_i (€)	19,007	17,599	18,536.55	18,383.46
C_o (€/year)	1304	440	272.576	292.7937
C_{oD} (€)	8012	2704	1674.86	1799.09
C_{tot} (€)	27,020	20,303	20,211	20,182

With all the equations described above, objective function is calculated by (41). Selected design variables for this minimization problem are tube outside diameter (d_o), shell diameter (D_s), baffles spacing (B) and number of tube passes (n). The lower and upper bounds for these variables are shown in Table 7

Discounted operating cost values are calculated with $ny = 10$ years, annual discount rate $i = 10\%$, energy cost $C_E = 0.12$ €/kW h and an annual amount of work hours $H = 7000$ h/year. These values are also used in [10,18,20,66]

5. Results and discussion

The effectiveness and robustness of the Improved Intelligent Tuned Harmony Search algorithm are tested with different cases studies acquired by the literature. These are namely

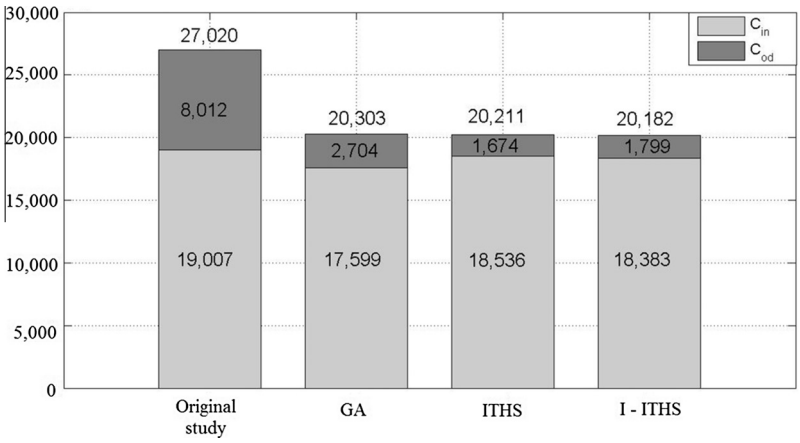

Figure 7 Cost comparison for case study 2.

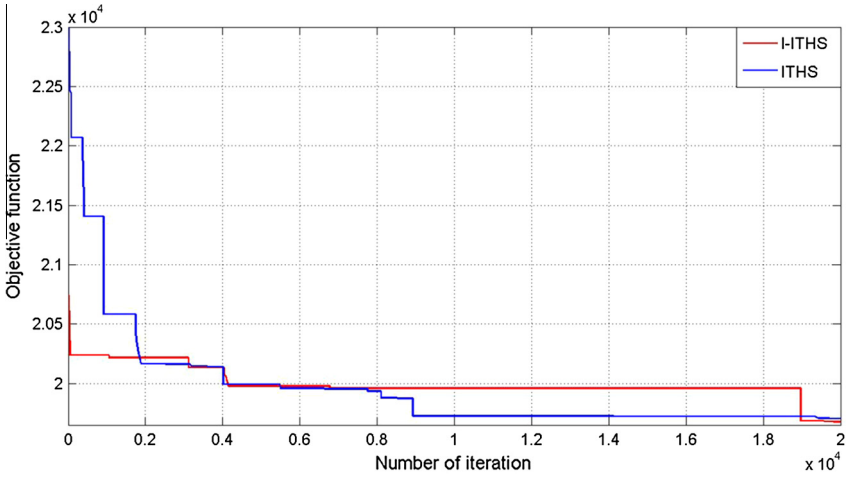

Figure 8 Convergence of ITHS for case study 2.

- Case 1: 4.34 (MW) heat duty, methanol–brackish water heat exchanger.
- Case 2: 1.44 (MW) duty, kerosene–crude oil heat exchanger.
- Case 3: 0.46 (MW) heat duty, distilled water-raw water heat exchanger.

The first case is taken from [65] and is related with a heat exchanger which has a heat duty of 4.34 MW. Heat exchange occurs between methanol and brackish water. Exchanger has a triangle pitch pattern and one shell side passage. Second case was firstly proposed in [65] and considered an exchanger with a square pitch pattern and one shell side passages. Heat exchange takes place between kerosene and crude-oil in an exchanger with a heat load of 1.44 MW. The third case was first discussed by Kern [65]. A heat exchanger with a 0.46 MW heat load with a square pitch pattern and one shell side passage is assumed. The values in Table 8 are specified as an input parameter for three cases. I-ITHS algorithm is adapted to solve these three cases and compared with the other algorithm's results. For a reliable comparison, the objective function for all three cases is represented with (41). Again for the consistency of the comparison, all the values related to cost are taken from Caputo et al. [10] who utilizes GA approach. ITHS algorithm is evaluated for 100 times by applying these algorithm parameters

- Harmony Memory Size = 25.
- Harmony Memory Considering Rate = 0.98.
- Maximum iteration = 40,000.

5.1. Case 1: 4.34 (MW) heat duty, methanol–brackish water heat exchanger

In Table 9, performance of the ITHS algorithm is compared with the other algorithms that were studied before. As it is seen, ITHS gives better results than the others. In this case all the analysis is made with the results gained from Sinnot [66]. Reduction in heat exchanger area is (18.06%) caused by reduction in tube length (61%) which is a function of number of tubes. Number of tubes increased significantly (287%) however, outside diameter of the tubes decreased. Due to the reduction of shell side velocity by 15.5% shell side pressure drop is decreased (58.2%). Adding to marked decrease in tube side pressure drop (%10.18), total pumping cost is significantly reduced (54.41%). The reduction of capital investment and

Table 11 Comparison of the algorithms for case study 3.

	Original study [41]	GA [10]	PSO [18]	ITHS	I-ITHS
D_s (m)	0.387	0.62	0.0181	0.5726	0.5671
L (m)	4.880	1.548	1.45	0.9737	0.9761
B (m)	0.305	0.440	0.423	0.4974	0.4989
d_o (m)	0.019	0.016	0.0145	0.0101	0.0100
P_t (m)	0.023	0.020	0.0187	0.0126	0.0125
C_l (m)	0.004	0.004	0.0036	0.0025	0.0025
n	2.0	2.0	2.0	2.0	2.0
N_t	160	803	894	1845	1846
v_t (m/s)	1.76	0.68	0.74	0.747	0.761
Re_t	36,409	9487	9424	6552	6614
Pr_t	6.2	6.2	6.2	6.2	6.2
h_t (W/m^2 K)	6558	6043	5618	5441	5536
f_t	0.023	0.031	0.0314	0.0369	0.0368
ΔP_t (Pa)	62,812	3673	4474	3869	4049
a_s (m^2)	0.0236	0.0541	0.059	0.0569	0.0565
D_e (m)	0.013	0.015	0.010	0.0071	0.0071
v_s (m/s)	0.94	0.41	0.375	0.3893	0.3919
Re_s	16,200	8039	4814	3473	3461
Pr_s	5.4	5.4	5.4	5.4	5.4
h_s (W/m^2 K)	5735	3476	4088.3	4832	4871
f_s	0.337	0.374	0.403	0.4238	0.4241
ΔP_s (Pa)	67,684	4365	4271	4995	5062
U (W/m^2 K)	1471	1121	1177	1220	1229
S (m^2)	46.6	62.5	59.2	57.3	56.64
C_i (€)	16,549	19,163	18,614	18,273	18,209
C_o (€/year)	4466	272	276	231	238
C_{oD} (€)	27,440	1671	1696	1419	1464
C_{tot} (€)	43,989	20,834	20,310	19,693	19,674

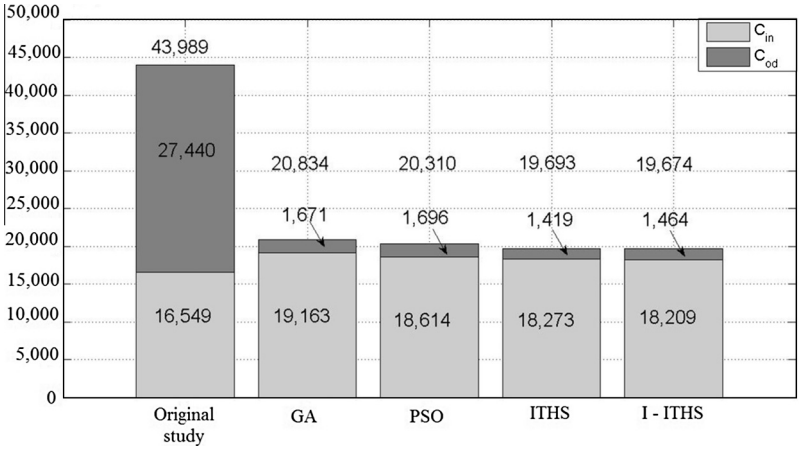

Figure 9 Cost comparison for case study 3.

pumping cost led to total discount of overall cost by 22.19%. An overall cost comparison is maintained in Fig. 5. Convergence of the Improved Intelligent Tuned Harmony Search (I-ITHS) and ITHS algorithms for case study 1 is implemented in Fig 6. As seen in Fig 6, problem is converged to minimum within 15,172 iterations by I-ITHS algorithm.

5.2. Case 2: 1.44 (MW) duty, kerosene–crude oil heat exchanger

In Table 10, performance of the I-ITHS algorithm is tested with the results obtained in literature survey. It is clearly seen that, the results gathered from I-ITHS are better than the original design which is implemented by Kern [65]. In this case, due to the increase of overall heat transfer coefficient (4.42%), heat exchanger area is reduced by (5.8%). And also there is a strong increase in numbers of tubes (95.7%) which is related by the ratio between D_s and d_o as seen in (31). Significant decrease in shell diameter (41.34%) accompanied with a slight increase tube length (3.62%). Therefore, capital investment which is a strong function of heat exchange area is decreased by 6.17%. Owing to the remarkable decrease of tube velocity, significant decrease (77.13%) in tube side pressure drop is seen. And also decrease in shell side flow velocity leads

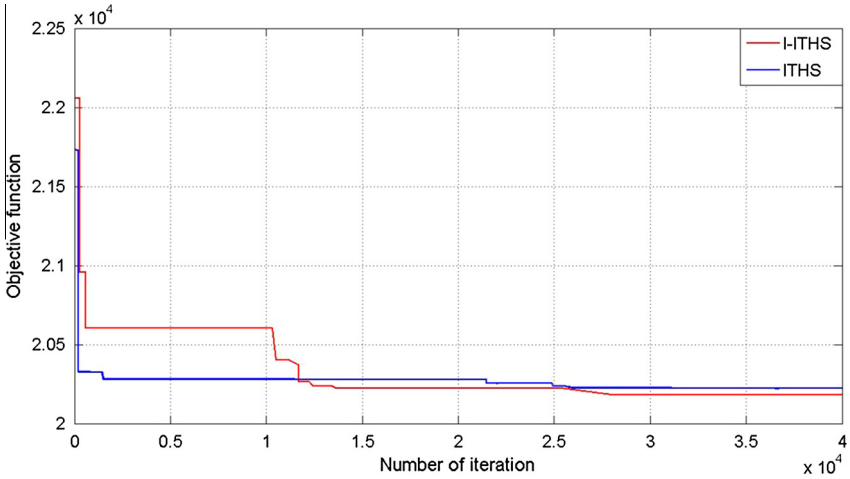

Figure 10 Convergence of ITHS for case study 3.

to considerable reduction (36.49%) in shell side pressure drop which causes sharp decrease (77.55%) in operational cost. Combined effect of operational and capital cost reduced the total cost of heat exchanger by 25.3%. Fig. 7 shows the total cost comparison of ITHS and original study taken by Kern's approach [65]. Fig. 8 details the convergence of the ITHS and I-THS algorithms. It is observed that I-ITHS algorithm converges to the minimum within 19,272 iterations.

5.3. Case 3: 0.46 (MW) heat duty, distilled water-raw water heat exchanger

I-ITHS algorithm is proposed for the case study 3 and compared with the other algorithms shown in Table 11. It is clearly observed that the results gained from I-ITHS algorithm are more satisfactory than the other algorithm's results. In this case, when compared to original design accomplished by Kern [65], due to the decrease (16.45%) of the overall heat transfer coefficient, an increase (21.55%) for the heat exchanger area is noticed. Therefore, a marked increase (9.9%) for capital investment cost is observed. Very sharp decrease (93.5%) in tube side pressure drop is seen due to the considerable reduction in tube side velocity (56.76%), and strong decrease (80.03%) in tube length. And also, thanks to sharp decrease (58.31%) of shell side flow velocity, decrease in tube length and increase (63.12%) in baffle spacing lead to very sharp decrease (92%) in shell side pressure drop. Total reduction of tube and shell side pressure drops led to critical decrease (94%) in operational cost values. This crucial reduction in capital and operational cost values led to strong decrease (55.28%) in total annual cost of heat exchanger.

Fig. 9 gives the total cost comparison for the case study 3. As shown in Fig. 9, ITHS algorithm gives more satisfactory results than the other algorithms. Fig. 10 shows the convergence performance of the ITHS algorithm for case study 3. As seen, Algorithm converges to minimum within 28,713 iterations.

6. Conclusion

Heat exchanger design is a complex task as there has been lot of studies devoted to this research area. Many advanced optimizing tools were adapted to specify successful design of heat exchangers. In this article, Improved Intelligent Tuned Harmony Search algorithm (I-ITHS) is proposed for cost minimization of shell and tube exchanger. I-ITHS algorithm, which has a few control parameters and easy in implementation, adopts the global search behavior of the Artificial Bee Colony algorithm and convergence speed of the proposed algorithm has been increased by the Opposition Based learning procedure. Several benchmark functions have been applied on I-ITHS and other Harmony search based variants. Numerical experiments revealed that I-ITHS algorithm surpasses other algorithms in terms of convergence speed and solution accuracy. The performance of I-ITHS algorithm is assessed with complex real world optimization problem related to optimum modeling of shell and tube heat exchanger. I-ITHS algorithm is applied to three case studies taken from literature. The results gathered from literature are compared with I-ITHS algorithm's findings. In all three cases, total cost reduction is observed with a less computational time. This feature displays the importance of the I-ITHS algorithm for engineering design problems. This paper shows that, manufacturers and designers can utilize Improved Intelligent Tuned Harmony Search algorithm for designing thermal systems and optimizing heat exchanger problems.

References

[1] Selbaş R, Kızılkan O, Reppich M. A new design approach for shell-and-tube heat exchangers using genetic algorithms from economic point of view. Chem Eng Process 2006;45:268–75.
[2] Minton PE. Process heat transfer. In: Proc 9th int heat transfer conf, heat transfer – Jerusalem. Paper no. KN-22, I; 1990. p. 355–62.
[3] Ponce-Ortega JM, Serna-Gonzalez M, Jimenez Gutierrez A. Use of genetic algorithms for the optimal design of shell-and-tube heat exchangers. Appl Therm. Eng. 2009;29:203–9.
[4] Wang Q, Chen Q, Chen G, Zeng M. Numerical investigation on combined multiple shell-pass shell-and-tube heat exchanger with continuous helical baffles. Int J Heat Mass Transfer 2009;52:1214–22.
[5] Fettaka S, Thibault J, Gupta Y. Design of shell-and-tube heat exchangers using multi-objective optimization. Int J Heat Mass Transfer 2013;60:343–54.
[6] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. IEEE Congr Evol Comput 2007:4661–7.

Please cite this article in press as: Turgut OE et al., Design and economic investigation of shell and tube heat exchangers using Improved Intelligent Tuned Harmony Search algorithm, Ain Shams Eng J (2014), http://dx.doi.org/10.1016/j.asej.2014.05.007

[7] Hadidi A, Hadidi M, Nazari A. A new design approach for shell-and-tube heat exchangers using imperialist competitive algorithm (ICA) from economic point of view. Energy Convers Manage 2013;67:66–74.
[8] Chaudhuri PD, Diwekar UM, Logsdon JS. An automated approach for the optimal design of heat exchangers. Ind Eng Chem Res 2007;36:3685–93.
[9] Mizutani FT, Pessoa FLP, Queiroz EM, Hauan S, Grossmann IE. Mathematical programming model for heat exchanger network synthesis including detailed heat exchanger designs. 1. Shell and tube heat exchanger design. Ind Eng Chem Res 2003;42:4009–18.
[10] Caputo AC, Pelagagge PM, Salini P. Heat exchanger design based on economic optimization. Appl Therm Eng 2008;28:1151–9.
[11] Özkol I, Komurgoz G. Determination of the optimum geometry of the heat exchanger body via genetic algorithm. Int J Heat Mass Transfer 2005;48:283–96.
[12] Hilbert R, Janiga G, Baron R, Thevenin D. Multiobjective shape optimization of a heat exchanger using parallel genetic algorithm. Int J Heat Mass Transfer 2006;49:2567–77.
[13] Xie GN, Sunden B, Wang QW. Optimization in calculation of shell and tube heat exchanger by a genetic algorithm. Appl Therm Eng 2008;28:895–906.
[14] Sun S, Lu Y, Yan C. Optimization in calculation of shell and tube heat exchanger. Int Commun Heat Mass Transfer 1993;20:675–85.
[15] Costa ALH, Queiroz EM. Design optimization of shell and tube heat exchanger. Appl Therm Eng, vol. 28. p. 1798–805.
[16] Wildi-Tremblay P, Gosselin L. Minimizing shell and tube heat exchanger cost with genetic algorithms and considering maintenance. Int J Energy Res 2007;31:867–85.
[17] Babu BV, Munawar SA. Differential evolution strategies for optimal design shell and tube heat exchangers. Chem Eng Sci 2007;62:3720–39.
[18] Patel VK, Rao RV. Design optimization of shell-and-tube heat exchanger using particle swarm optimization technique. Appl Therm Eng 2010;30:1417–25.
[19] Sahin AS, Kilic B, Kilic U. Design and economic optimization of shell and tube heat exchangers using Artificial Bee Colony (ABC) algorithm. Energy Convers Manage 2011;52:3356–62.
[20] Mariani VC, Duck ARK, Guerra FA, Coelho LdS, Rao RV. A chaotic quantum behaved particle swarm approach applied to optimization of heat exchangers. Appl Therm Eng 2012;42:119–28.
[21] Hadidi A, Nazari A. Design and economic optimization of shell and tube heat exchangers using biogeography-based (BBO) algorithm, vol. 51; 2013. p. 1263–72.
[22] Yadav P, Kumar R, Panda SK, Chang CS. An Intelligent tuned harmony search algorithm for optimisation. Inform Sci 2012;196:47–72.
[23] Henon M. A two-dimensional mapping with a strange attractor. Commun Math Phys 1976;50:69–77.
[24] Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department; 2005.
[25] Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. In: Proc int conf comput intell modelling control and autom, vol. 1; 2005. p. 695–01.
[26] Geem ZW, Kim JH, Loganathan GV. A new metaheuristic optimization algorithm: harmony search. Simulation; 2001.
[27] Saka MP. Optimum design of steel sway frames to BS5950 using harmony search algorithm. J Constr Steel Res 2009;65:36–43.
[28] Erdal F, Dogan E, Saka MP. Optimum design of cellular beams using harmony search and particle swarm optimizers. J Constr Steel Res 2011;67:237–47.
[29] Sivasubramani S, Swarup KS. Multi-objective harmony search algorithm for optimal power flow problem. Int J Electr Power Energy Syst 2011;33:745–52.
[30] Vasebi A, Fesanghary M, Bathaeea SMT. Combined heat and power economic dispatch by harmony search algorithm. Int J Electr Power Energy Syst 2007;29:713–9.
[31] Sivasubramani S, Swarup KS. Environmental/economic dispatch using multi-objective harmony search algorithm. Electr Power Syst Res 2011;81:1778–85.
[32] Manjarres D, Del Ser J, Gil-Lopez S, Vecchio M, Landa-Torres I, Salcedo-Sanz S, Lopez-Valcerce R. A novel heuristic approach for distance and connectivity based multihop node localization in wireless sensor networks. Appl Soft Comput 2012:1–12.
[33] Manjarres D, Del Ser J, Gil-Lopez S, Vecchio M, Landa-Torres I, Salcedo-Sanz S, Lopez-Valcerce R. On the design of a novel two objective harmony search approach for distance and connectivity based location in wireless sensor networks. Eng Appl Artif Intell 2013;26:669–76.
[34] Landa-Torres I, Gil-Lopez S, Del Ser J, Salcedo-Sanz S, Manjarrez D, Portilla-Figueras JA. Efficient citywide planning of open WiFi access networks using novel grouping harmony search heuristics. Eng Appl Artif Intell 2012;26:1124–30.
[35] Yazdi E, Azizi V, Haghighat AT. A new biped locomotion involving arms swing based on neural network with harmony search optimizer. In: IEEE international conference on automation and logistics; 2011. p. 18–23.
[36] Das Sharma K, Chatterjee A, Rakshit A. Design of a hybrid stable adaptive fuzzy controller employing lyapunov theory and harmony search algorithm. IEEE Trans Control Syst Technol 2011;18:1440–7.
[37] Panchal A. Harmony search in therapeutic medical physics. Stud Comput Intell 2009;191:189–203.
[38] Panchal A. Harmony search optimization for HDR prostate brachytherapy. Rosalind Franklin University of Medicine and Science; 2008.
[39] Gandhi TK, Chakraborty P, Roy GG, Panigrahi BK. Discrete harmony search based expert model for epileptic seizure detection in electroencephalography. Exp Syst Appl 2012;39:4055–63.
[40] Geem ZW. Optimal cost design of water distribution network using harmony search. Eng Optim 2006;38:259–80.
[41] Gao XZ, Wang X, Ovaska SJ, Zenger K. A hybrid optimization method of harmony search and opposition-based learning. Eng Optim 2012;44:895–914.
[42] Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 2005;194:3902–33.
[43] Wang CM, Huang YF. Self-adaptive harmony search algorithm for optimization. Exp Syst Appl 2010;37:2826–37.
[44] Alatas B. Chaotic harmony search algorithms. Appl Math Comput 2010;216:2687–99.
[45] Xiang W-I, An M-Q, Li Y-Z, He R-C, Zhang J-F. An improved global-best harmony search algorithm for faster optimization. Exp Syst Appl 2014.
[46] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. Appl Math Comput 2007;188:1567–79.
[47] Omran MGH, Mahdavi M. Global-best harmony search. Appl Math Comput 2008;198:643–56.
[48] Zou D, Gao L, Wu J, Li S. Novel global harmony search algorithm for unconstrained problems. Neurocomputing 2010;73:3308–18.
[49] Jaberipour M, Khorram E. Two improved harmony search algorithms for solving engineering optimization problems. Commun Nonlinear Sci Numer Simul 2010;15:3316–31.
[50] Ashrafi SM, Dariane AB. Performance evaluation of an improved harmony search algorithm for numerical optimization: Melody Search (MS). Eng Appl Artif Intell 2013;26:1301–21.
[51] Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition – based differential evolution algorithms. In: IEEE congress on evolutionary computation, 16–21 July, Vancouver, Canada; 2006. p. 2010–7.

Please cite this article in press as: Turgut OE et al., Design and economic investigation of shell and tube heat exchangers using Improved Intelligent Tuned Harmony Search algorithm, Ain Shams Eng J (2014), http://dx.doi.org/10.1016/j.asej.2014.05.007

[52] Han L, He XS. A novel opposition based particle swarm optimization for noisy problems. In: International conference on the applications on natural computation, 24–27 August, Haikou, China; 2007. p. 624–9.
[53] Malisia AR, Tizhoosh HR. Applying opposition-based ideas to the ant colony system. In: IEEE swarm intelligence symposium, 1–5 April, Honolulu, USA; 2007. p. 182–9.
[54] El-Abd M. Opposition – based artificial bee colony algorithm. In: 13th Annual conference on genetic and evolutionary computation, 12–16 July, Dublin, Ireland; 2011. p. 109–15.
[55] Rahnamayan S, Tizhoosh HR, Salama MMA. Quasi – oppositional differential evolution. In: IEEE congress on evolutionary computation, 25–28 September, Singapore; 2007. p. 2229–36.
[56] Wang H, Wu ZJ, Liu Y, Wang J, Jiang DZ, Chen LL. Space transformation search: a new evolutionary technique. In: ACM/SIGEVO summit on genetic and evolutionary computation, 12–14 June, Shanghai, China; 2009. p. 537–44.
[57] Tishoosh HR. Reinforcement learning model based on actions and opposite actions. In: Proc ICGST in conf artif intel mach learn. Egypt; 2005.
[58] Liu B, Wang L, Jin YH, Tang F, Huang DX. Directing orbits of chaotic systems by particle swarm optimization. Chaos, Soliton Fract 2006;29:454–61.
[59] Ott E. Chaos in dynamical systems. Cambridge (UK): Cambridge University Press; 2002.
[60] May RM. Simple mathematical models with very complicated dynamics. Nature 1976;261:459–67.
[61] He D, He C, Jiang L, Zhu H, Hu G. Chaotic characteristic of a one dimensional iterative map with infinite collapses. IEEE Trans Circ Syst 2001;48:900–6.
[62] Ahmadi M, Mojallali H. Chaotic invasive weed optimization algorithm with application to parameter estimation of chaotic systems. Chaos, Soliton Fract 2012;45:1108–20.
[63] Dong N, Wu C-H, Ip W-H, Chen Z-Q, Chan C-Y, Yung K-L. An opposition based chaotic GA/PSO hybrid algorithm and its application in circle detection. Comput Math Appl 2012;64:1886–902.
[64] He Y, Yang S, Xu Q. Short-term cascaded hydro electric system scheduling based on chaotic particle swarm optimization using improved logistic map. Communications in Nonlinear Science and Numerical Simulation 2013;18(7):1746–56.
[65] Kern DQ. Process heat transfer. New York: McGraw-Hill; 1950.
[66] Sinnot RK. Coulson and Richardson's chemical engineering. Chemical engineering design, vol. 6. Butterworth-Heinemann; 2005.
[67] Hewitt GF. Heat exchanger design handbook. New York: Begell House; 1998.
[68] Shah RK, Bell KJ. Handbook of thermal engineering. Florida: CRC Press; 2000.
[69] Serth RW. Process heat transfer – principles and applications. Elsevier Science & Technology Books; 2007.
[70] Rosenhow WM, Harnett PJ. Handbook of heat transfer. New York: McGraw-Hill; 1973.
[71] Fraas AP. Heat exchanger design. 2nd ed. New York: John Wiley; 1989.
[72] Peters MS, Timmerhaus KD. Plant design and economics for chemical engineers. New York: McGraw-Hill; 1991.
[73] Taal M, Bulatov I, Klemes P, Stehlik P. Cost estimation and energy price forecast for economic evaluation of retrofit project. Appl Therm Eng 2003;23:1819–35.

Oguz Emrah Turgut was born in Polatlı, Ankara. He received his B.Sc. from DokuzEylül University in 2008 and M.Sc. from Ege University in 2011. He is now Ph.D. student in Energy Department, in Ege University. His research interest are Thermodynamic, Heat Transfer and Computer Science.

Mert Sinan Turgut received his B.Sc. on Mechanical Engineering Department in Dokuz Eylul University in 2010. He is a M.Sc. student on Mechatronics Engineering in Dokuz Eylul University. His research interests include control theory, inverse problems and optimization algorithms.

Mustafa Turhan Coban, borned in Bolu, Seben, Turkey in 1957. He received his B.Sc. degree from Ege University School of Mechanical Engineering, Department of Mechanical Engineering in 1978, and his M.Sc. degree in Mechanical Engineering from Michigan Technological University (U.S.A.) in 1982. He graduated from University of Utah (U.S.A.) in 1986 with a Ph.D. in Mechanical Engineering. In 1995, he received Postgraduate degree in Computer Science from Victoria Technological University (Australia). He is now assistant professor in Ege University. His research interests are fuel cell technologies, air conditioning and computer science.

Please cite this article in press as: Turgut OE et al., Design and economic investigation of shell and tube heat exchangers using Improved Intelligent Tuned Harmony Search algorithm, Ain Shams Eng J (2014), http://dx.doi.org/10.1016/j.asej.2014.05.007