

The Design of the Best Heat Integrated Separation Systems Using Harmony Search Algorithm

Kazem Hasanzadeh Lashkajani¹, Bahram Ghorbani^{2*},
Gholam Reza Salehi³ and Majid Amidpour⁴

^{1,2,4} Energy System Engineering Department, K.N.Toosi University of Technology, Iran

³Islamic Azad University Branch of Nowshahr, Iran

(Received: 21 July 2012, Accepted: 15 November 2012)

Abstract: The synthesis of heat integrated multi-component distillation systems is complex due to its huge search space for structural combination and optimization computation. To provide a systematic approach and tools for the synthesis design of distillation systems, a new method for modeling heat integrated columns is presented, and the operating cost objective function is minimized by improved harmony search algorithm (IHS). This paper studies a quick method for the synthesis of heat integrated distillation column sequences and IHS -based optimization strategy for the optimization of separation sequences with their heat integration.

Keywords: Distillation Sequence, Heat Integration, Optimization, IHSA

1. Introduction

The synthesis design of heat integrated distillation systems is a major challenge in the area of computer aided design. Despite all the efforts invested in the area, practical system synthesis and integration still reside in the domain of human experts. Due to the inherent uniqueness and complexity of each new process, a systematic and comprehensive approach for automating process design has remained elusive. The synthesis of heat integrated distillation systems will, almost by definition, require optimization of the following three interrelated aspects: (1) determining the sequence of units with their interconnections; (2) designing each unit in the sequence and suitable operating condition, e.g. operating pressure for a distillation column; and (3) designing the heat exchanger network.

* Corresponding Author.

Authors' Email Address: ¹{k.hasanzadeh@yahoo.com}, ²{bahram330ghorbani@gmail.com}, ³{rezasalehi20@gmail.com},

⁴{amidpour@kntu.ac.ir}

Significant advances have been made in these three aspects in the last 30 years. Heuristic methods are proposed by a number of researchers (Liu 1987; Andreacovich & Westerberg, 1985). Indeed, heuristic rules play an important role in reducing the huge search space when coupled with heat integration. However, the lack of complete and systematic knowledge required often causes difficulty and leads to low quality solutions. Qian & Lien (1994) proposed a fuzzy logic-based heuristic rule match approach to synthesis complex separation sequences, extending to distillation and extraction operations. This work is a semi-quantitative screening procedure to subtract the original large hyper structure into a much smaller superstructure for further detailed optimization computation. Decomposition approaches decompose the synthesis problem into two major steps: separation sequencing and synthesis of the heat exchange network. The idea of decomposition arises from an assumption that near-optimal heat in-integrated rectification sequences tends to possess the largest potential for heat integration (Linnhoff, 1982). The method is used in conjunction with pinch technology (Linnhoff, 1983; Lucia & McCallum, 2010) that has shown to be a reliable and effective tool for finding minimum energy distillation column sequences. Although pinch point technology has been proved to be very successful in engineering practice, it is primarily used for analyzing flow sheets, not for synthesizing. The best rectification system, however, cannot be determined by separation characteristic alone. A heat exchanger network must be considered and all parts in the system are in mutual interaction. Thus, the decomposition method cannot guarantee a global optimum, and pinch technology does not suffice for the synthesis of energy integrated distillation sequences. Mixed integer linear or non-linear programming methods (MIL/NLP) have been suggested by a number of researchers (Giridhar & Agrawal, 2010; Kravanja & Grossmann 1990; Zorka, 1994; Zdravko & Grossmann, 1997; Proios et al. 2005). Because of the complexity and high computational expense of standard mathematical programming, such as branch-and-bound method, generalized benders decomposition method (Proios et al. 2005) and outer-approximation algorithm (OA), many simplifications have to be assumed to make the mathematical models manageable, which rend the approach severe limitations in practical applications. Although recent advances exhibit a considerable degree of heat integration into model formulation, a simultaneous optimization method has not been developed in the way which would allow efficient flow-sheet optimization.

Mathematically, synthesis and optimization of heat integrated distillation systems are to search for the maximum in a mixed integer and continuous variables space. Due to the non-monotonic and local maximal properties of the search space, conventional derivative-based optimization algorithms turned out to be incapable of finding the global maximal in most cases. Random search algorithms are capable of the multi-modal optimization problem. Since they are known to be extremely computationally expensive, random search algorithms were not widely accepted in engineering applications until computing technology became affordable and fast enough in recent years. Among random algorithms, harmony search algorithms have shown their merits in large-scale parallelism search, fast and steady approaching to global optimal (Lee & Geem, 2005). However, there has been no report of application of HSA for synthesis and optimization of heat integrated separation sequences and heat exchanger network. In this work, authors present a new algorithm for calculation of heat

integration and the objective function of the cost of sequences using an improved harmony search algorithm (IHS) to challenge the task.

2. Improved harmony search algorithm

This section describes the proposed improved harmony search (IHS) algorithm. A brief overview of the HS and the modification procedures of the IHS algorithm are provided.

2.1. Harmony search algorithm

Harmony search (HS) algorithm was recently developed in an analogy with music improvisation process where music players improvise the pitches of their instruments to obtain better harmony (Lee, 2004). The steps in the procedure of harmony search are shown in Figure 1. They are as follows:

- Step 1. Initialize the problem and algorithm parameters.
 - Step 2. Initialize the harmony memory.
 - Step 3. Improvise a new harmony.
 - Step 4. Update the harmony memory.
 - Step 5. Check the stopping criterion.
- These steps are described in the next five subsections.

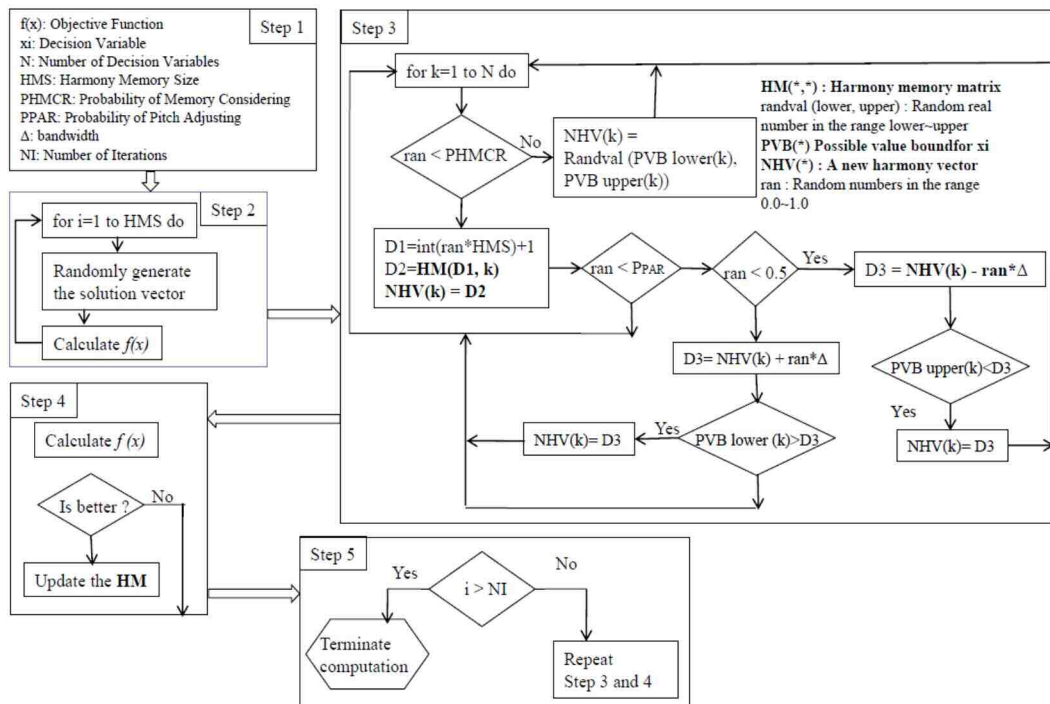


Fig. 1. Optimization procedure of the improved harmony search algorithm (Mahdavi, 2007)

2.1.1. Initialize the problem and algorithm parameters

In Step 1, the optimization problem is specified as follows:

$$\text{Minimize } f(x) \text{ subject to } x_i \in X_i, i=1,2,\dots,N \quad (1)$$

Where $f(x)$ is an objective function; x is the set of each decision variable x_i ; N is the number of decision variables, X_i is the set of the possible range of values for each decision variable, that is ${}_L X_i \leq X_i \leq {}_U X_i$; and ${}_L X_i$ and ${}_U X_i$ are the lower and upper bounds for each decision variable. The HS algorithm parameters are also specified in this step. These are the harmony memory size (HMS) or the number of solution vectors in the harmony memory, harmony memory considering rate (HMCR), pitch adjusting rate (PAR), and the number of improvisations (NI) or stopping criterion.

The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored. This HM is similar to the genetic pool in the GA (Geem & Loganathan 2002). Here, HMCR and PAR are parameters that are used to improve the solution vector. Both are defined in Step 3.

2.1.2. Initialize the harmony memory

In Step 2, the HM matrix is filled with as many randomly generated solution vectors as the HMS

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \dots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \dots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix} \quad (2)$$

2.1.3. Improvise a new harmony

A new harmony vector, $x' = (x'_1, x'_2, \dots, x'_N)$, is generated based on three rules: (1) memory consideration, (2) pitch adjustment and (3) random selection. Generating a new harmony is called 'improvisation' (Lee & Geem, 2005).

In the memory consideration, the value of the first decision variable (x'_1) for the new vector is chosen from any of the values in the specified HM range ($x_1^1 - x_1^{HMS}$). Values of the other decision variables ($x'_2, x'_3, \dots, x'_N$) are chosen in the same manner. The HMCR, which varies between 0 and 1, is the rate of choosing one value from the historical values stored in the HM, while $(1-HMCR)$ is the rate of randomly selecting one value from the possible range of values.

$$x'_i \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & \text{with probability } HMCR \\ x_i' \in X_i & \text{with probability } (1 - HMCR) \end{cases} \quad (3)$$

For example, a HMCR of 0.85 indicates that the HS algorithm will choose the decision variable value from historically stored values in the HM with an 85 % probability or from the entire possible range with a (100–85) % probability. Every component obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the rate of pitch adjustment as follows:

Pitch adjusting decision for

$$x'_i \leftarrow \begin{cases} \text{Yes} & \text{With probability } PAR \\ \text{No} & \text{With probability } (1 - PAR) \end{cases} \quad (4)$$

The value of $(1 - PAR)$ sets the rate of doing nothing. If the pitch adjustment decision for x'_i is YES, x'_i is replaced as follow:

$$x'_i \leftarrow x'_i \pm \text{rand}() \times bw \quad (5)$$

where bw is an arbitrary distance bandwidth and $\text{rand}()$ is a random number between 0 and 1.

In Step 3, HM consideration, pitch adjustment or random selection is applied to each variable of the new harmony vector in turn.

2.1.4. Update harmony memory

If the new harmony vector, $x' = (x'_1, x'_2, \dots, x'_N)$ is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

2.1.5. Check stopping criterion

If the stopping criterion (maximum number of improvisations) is satisfied, computation is terminated. Otherwise, Steps 3 and 4 are repeated.

2.2. Improved parameters

The HMCR and PAR parameters, introduced in Step 3, help the algorithm to find globally and locally improved solutions, respectively (Lee & Geem 2005).

PAR and bw in HS algorithm are very important parameters in fine-tuning of optimized solution vectors, and can be potentially useful in adjusting convergence rate of algorithm to optimal solution. So, fine adjustment of these parameters is of great interest.

The traditional HS algorithm uses fixed value for both PAR and bw . In the HS method, PAR and bw values adjusted in initialization step (Step 1) cannot be changed during new generations. The main drawback of this method appears in the number of iterations that the algorithm needs to find an optimal solution.

Small PAR values with large bw values can cause poor performance of the algorithm and considerable increase in iterations needed to find optimum solution. Although small bw values in final generations increase the fine-tuning of solution vectors, but in early generations bw must take a bigger value to enforce the algorithm to increase the diversity of solution vectors. Furthermore, large PAR values with small bw values usually cause the

improvement of best solutions in final generations which algorithm converged to optimal solution vector.

The key difference between IHS and traditional HS method is in the way of adjusting PAR and bw. To improve the performance of the HS algorithm and eliminate the drawbacks which lie with fixed values of PAR and bw, IHS algorithm uses variables PAR and bw in improvisation step (Step 3). PAR and bw are changed dynamically with generation number as shown in Figure 2 and expressed as follow:

$$PAR(gn) = PAR_{\min} + \frac{(PAR_{\max} - PAR_{\min})}{NI} \times gn \quad (6)$$

where PAR is the pitch adjusting rate for each generation, $PAR_{\min}$ is the minimum pitch adjusting rate, $PAR_{\max}$ is the maximum pitch adjusting rate, NI is the number of solution vector generations, and gn is the generation number.

$$bw(gn) = bw_{\max} \exp\left(c \cdot gn\right) \quad (7)$$

$$c = \frac{\ln\left(\frac{bw_{\max}}{bw_{\min}}\right)}{NI}$$

where $bw(gn)$ is the bandwidth for each generation, $bw_{\min}$ is the minimum bandwidth, and $bw_{\max}$ is the maximum bandwidth.

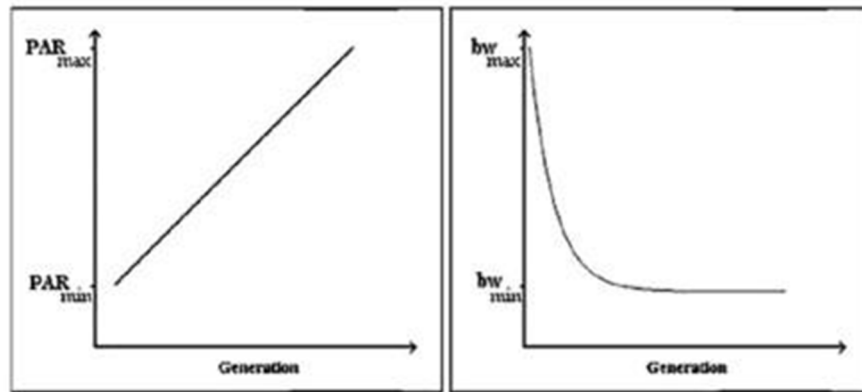


Fig. 2. Variation of PAR and bw versus generation number (Mahdavi, 2007)

3. Problem formulation by using coding representations

3.1. Problem statement

For the given N -component mixture of known conditions (composition, Flow rate, temperature and pressure), the problem is finding a distillation system for separating the mixture into N products corresponding to the components, with a flow sheet structure including separation sequence and heat integration and operating parameters that lead to the lowest operating cost. The assumptions made for the problem are as follows:

(1) Heating and cooling requirements for each column are not directly provided by hot and cold utilities; heat integration should be taken into consideration for economic-technological reasons.

(2) Pressures are selected from the range of allowable pressures. This allows column pressure to vary within an allowable range in order to carry out heat integration. Pressures of columns are treated as continuous variable.

(3) Each distillation column performs a simple split (i.e. one feed and two products) and sharp split.

(4) The minimum temperature difference for heat transfer between two streams is equal to 10 K. ($\Delta T_{\min} = 10$ K)

From the foregoing problem statement, the optimization problem has two kinds of decision variables: discrete decisions for the flow sheet structure (sequences) and continuous decisions for the individual columns' operations, the former of which indicate clearly the simple separation sequence structures and the heat integration configurations. In the system, if all the individual columns come up with their optimized operating conditions, the only continuous decision parameters to be optimized are the operating pressures $\{p\}$, which could be adjusted to alter the heat integrations. So, if the discrete decisions could be expressed as sequence discrete variable $\{s_i\}$, the optimization problem could be formulated as:

$$\text{Min COST} = \min C(\{s_i\}, \{p\}), \{s_i\} \in S, \{p\} \in P \quad (8)$$

where $C()$ represents the cost model for the system, S is a set including all possible flow sheet sequences, P is the vector of operating pressures for all individual columns in the system, and P can be a set of feasible range of $\{p\}$. The feasible range of pressure of a column is determined by the allowable pressure/temperature range for the column, which is related to the composition of the overhead and the bottom, the available hottest and coldest utility temperatures, as well as the critical pressure/temperature of the components in the mixture.

3.2. Coding approach for separation sequence representation

Consider an N -component mixture, and the components $\{C_1, C_2, C_i, C_{i+1}, \dots, C_N\}$ which are ordered according to their volatilities (i.e. C_1 is the most volatile and C_N the least). We use nature numbers 1, 2, $\dots$, $N-1$ as the codes to name the cuts between each pair of adjacent components, as shown in Figure 3.

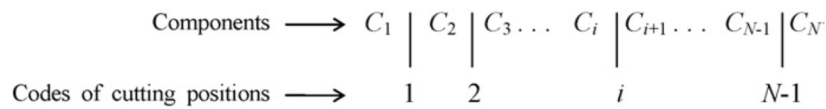


Fig. 3. Codes for naming the cutting positions

Typically, code i , indicates the split between components C_i and C_{i+1} , and its value is identical to the order of the light-key component C_i for the split. Thus, an ordered $(N-1)$ code sequence $\{s_i\}$ ($i=1, 2, \dots, N-1$) can be used to denote a separation sequence. For example, for a mixture of $N=6$, the sequence $\{s_i\} = \{2, 4, 5, 1, 3\}$ can represent the separation sequence $C_1C_2 | C_3C_4C_5C_6 \rightarrow C_3C_4 | C_5C_6 \rightarrow C_5 | C_6 \rightarrow C_1 | C_2 \rightarrow C_3 | C_4$.

This example indicates that a code (a cutting position) in Figure 3 can be used to denote anyone in the family of separation tasks that has the same pair of light and heavy key components. For example code “2” in Figure 3 can be used to represent the separation tasks $C_2 | C_3$, $C_1C_2 | C_3$, $C_1C_2 | C_3C_4$, or $C_1C_2 | C_3, \dots, C_N$, etc. The separation task represented by a code can be identified in sequence $\{s_i\}$ according to its position in the sequence. For example, the element “2” in sequence $\{s_i\}$ denotes the separation task $C_1C_2 | C_3, \dots, C_N$, as it is the first element in $\{s_i\}$. To manipulate code sequence $\{s_i\}$ and generate separation sequences, a method is needed to identify systematically the separation tasks denoted by the elements in $\{s_i\}$. In this work, a binary matrix approach is developed for this purpose. To represent a separation sequence, as demonstrated in Figure 4 for a six-component problem, we propose the following procedure:

Step 1. Generate a crude code series randomly by nature numbers $1, 2, \dots, N-1$ without the same number as a sequence (Figure 4 – Step 1).

Step 2. Generate a binary connection matrix structure for the sequence. Compare the new binary connection matrix structure with the binary connection matrix structures of sequences saved in the past. If it is not similar to them, go to Step 3 and if it is similar, go to step 1. The binary connection matrix structure demonstrates the connections among the nodes in the sequence. As presented in Figure 4 – Step 2, in the binary connection matrix, the row 2(node 2) is connected to columns 1(node 1) and 4(node 4). Also, the row 4(node 4) is connected to columns 3(node 3) and 5(node 5). This matrix always is unique for each sequence.

Step 3. Generate a binary task matrix structure for the sequence. The specifications of each stream in the sequence are extracted from the binary task matrix of the sequence. As demonstrated in Figure 4 – Step 3, in the binary task matrix, the row 1 and 2 demonstrate the task of node 2 in the sequence. The key components for the split, the specifications, compositions and flow rates of all streams in the node 2 can be calculated by the row 1 and 2. The row 3 and 4 demonstrate the task of node 4 in the sequence. The row 5 and 6 demonstrate the task of node 5 in the sequence. The row 7 and 8 demonstrate the task of node 1 in the sequence. And the row 9 and 10 demonstrate the task of node 3 in the sequence.

Step 4. Save the sequence, the binary connection matrix structure, and the binary task matrix structure.

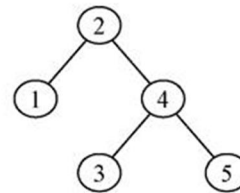
With such an identification algorithm, a unique separation sequence $\{s_i\}$ can be systematically generated, and with the corresponding specified recoveries of key components for each split, the specifications, compositions and flow rates of all streams in a separation network can be calculated.

Step 1: Generate a crude sequence

$\{2, 4, 5, 1, 3\}$

Step 2: Generate a binary connection matrix

	1	2	3	4	5
1	0	0	0	0	0
2	1	0	0	1	0
3	0	0	0	0	0
4	0	0	1	0	1
5	0	0	0	0	0



Step 3: Generate a binary task matrix and
Identify the separation tasks

	1	2	3	4	5	
1	1	0	0	0	0	$C_1C_2 C_3C_4C_5C_6$
2	0	0	1	1	1	$C_3C_4 C_5C_6$
3	0	0	1	1	0	$C_5 C_6$
4	0	0	0	0	1	$C_1 C_2$
5	0	0	0	0	0	$C_3 C_4$
6	1	0	0	0	0	
7	0	1	0	0	0	
8	0	0	1	0	0	
9	0	0	0	1	0	
10	0	0	0	0	1	

Step 4: Generate sequence $\{s_i\}$

$\{s_i\} = \{2, 1, 4, 3, 5\}$

Fig. 4. Separation sequence representation

3.3. Coding approach for heat integration configuration and objective function calculation

For each sequence, the compositions of feed stream, top products and bottom products of all $n-1$ columns are calculated. All units receive energy from the environment or give off energy to it. From the viewpoint of process energy utilization, condensers and reboilers can be treated as hot streams and cold streams, respectively. To represent heat integration and calculate the objective function of cost, we propose the following procedure (Figure 5):

Step 1. Select the sequence $\{s_i\}$ and column pressures $\{P\}$

Step 2. Calculate the temperature (T) and heat duties (Q) of condensers and reboilers of each column and make the matrix $[T \ Q]_{((2(n-1))*2)}$ which includes condenser temperatures and duties from row 1 to $(n-1)$ and reboiler temperatures plus $\Delta T_{\min}$ and duties from row n to $(2(n-1))$. Q is positive for reboilers and is negative for condensers.

Step 3. Sort the column of T in the [T Q] matrix, from the highest temperature to lowest temperature and arrange the column of Q according to the positions of sorted T. This sorting shows which streams as heat source can transfer heat to other stream as heat sinks. In next steps the heat integration and the cost of hot and cold utilities will be calculated.

Step 4. For $i = 1$ to $2(n-1)$ calculate $H = Q + Q_m$. Now, from maximum temperature to minimum temperature, we start calculating the hot utility consumption and their heat flow rates and temperatures.

Step 5. If $H > 0$ calculate the cost of hot utilities according to the temperature of the stream and available hot utility and the amount of H. place $Q_m = 0$. End if.

Step 6. If $H < 0$ place $Q_m = H$. End if.

Step 7. End for and calculate the total cost of hot utilities.

Step 8. For $i = 2(n-1)$ to 1 calculate $H = Q + Q_m$. Now, from minimum temperature to maximum temperature, we start to calculate cold utility consumption and their heat flow rates and temperatures.

Step 9. If $H < 0$ calculate the cost of cold utilities according to the temperature of the stream and available cold utility and the amount of H. place $Q_m = 0$. End if.

Step 10. If $H > 0$ place $Q_m = H$. End if.

Step 11. End for and calculate the total cost of cold utilities.

Step 12. For $i = 1$ to $2(n-1)$. Now, from maximum temperature to minimum temperature, we start to calculate the heat integration, stream temperatures and heat transferred among them.

Step 13. If $Q < 0$ save temperature T and Heat flow rate Q in a matrix $COL = [T \ Q]$ and calculate $HC = HC + Q$. End if.

Step 14. If $Q > 0$ & $HC < 0$ place $Q^* = Q$.

Step 15. For $j = 1: m$ ($m =$ number of rows in COL matrix) calculate $HI = COL(j, 2) + Q^*$.

Step 16. If $HI > 0$ calculate $QR = COL(j, 2) + QR$.

Step 17. Save temperature $COL(j, 1)$ as hot stream, temperature $T - \Delta T_{min}$ as cold stream and heat flow rate $COL(j, 2)$ as heat transferred between them, in matrix $ThTcQ = [COL(j, 1) \ T - \Delta T_{min} \ COL(j, 2)]$.

Step 18. Delete row $COL(j, 2)$ in COL matrix and place $Q^* = HI$. End if.

Step 19. If $HI < 0$ calculate $QR = -(Q + QR)$.

Step 20. Save temperature $COL(j, 1)$ as hot stream, temperature $T - \Delta T_{min}$ as cold stream and heat flow rate $COL(j, 2)$ as heat transferred between them, in matrix $ThTcQ = [COL(j, 1) \ T - \Delta T_{min} \ COL(j, 2)]$.

Step 21. Place $Q^* = 0$, $QR = 0$, $COL(j, 2) = HI$. End for j.

Step 22. Calculate $HC = HC + Q$.

Step 23. If $HC > 0$ place $HC = 0$. End if.

Step 24. End for and save the matrix ThTcQ (heat integration data).

Step 25. End and calculate the sum of hot utility cost and cold utility cost as total operating cost.

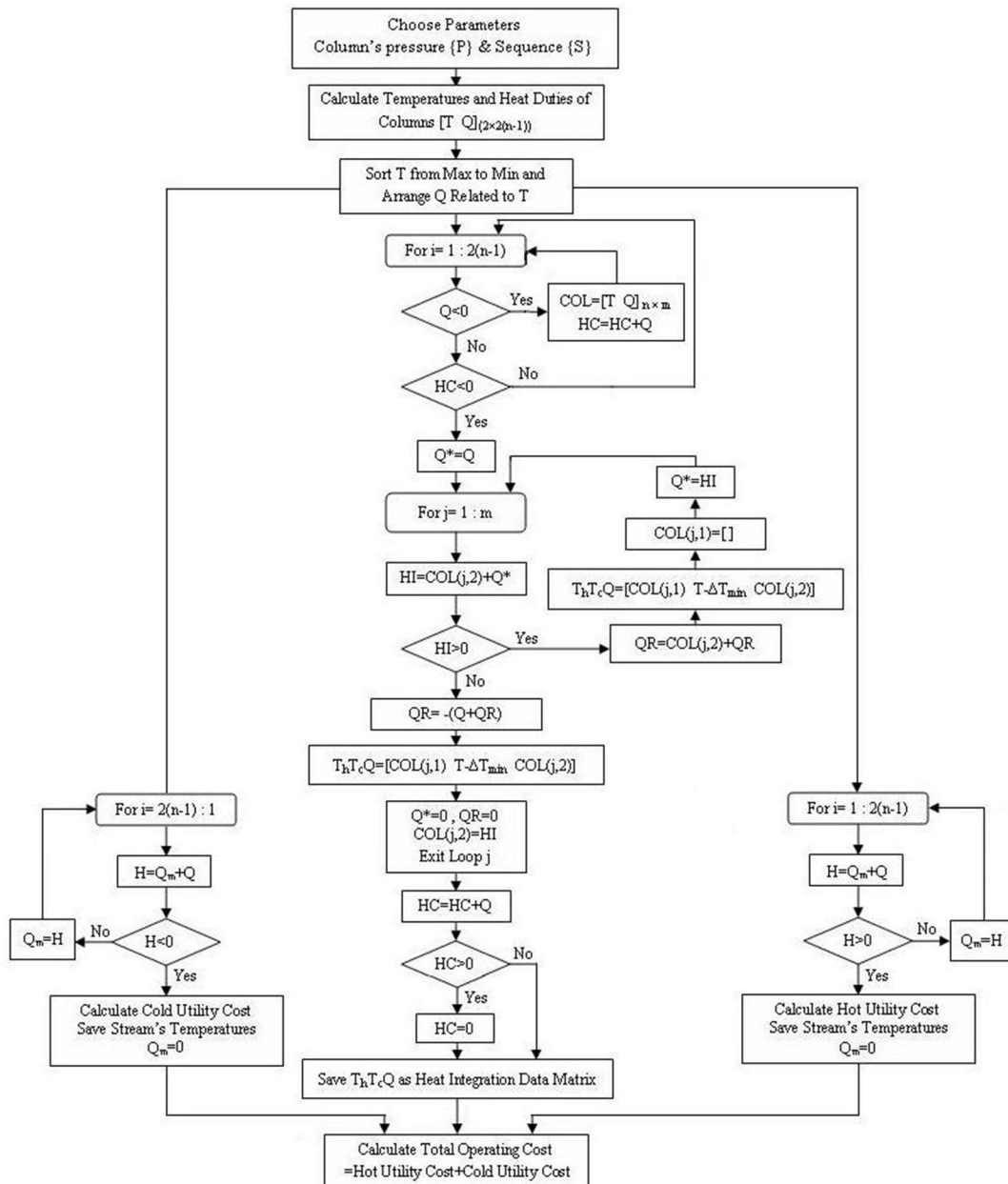


Fig. 5. Presented algorithm for calculating total operating cost and heat integration of sequences

A separation sequence with four columns is presented on T- ΔH diagram to show the calculation direction of proposed algorithm (Figure 6). Figure 6 shows the directions of calculations of hot utilities, cold utilities and heat integration by the proposed algorithm.

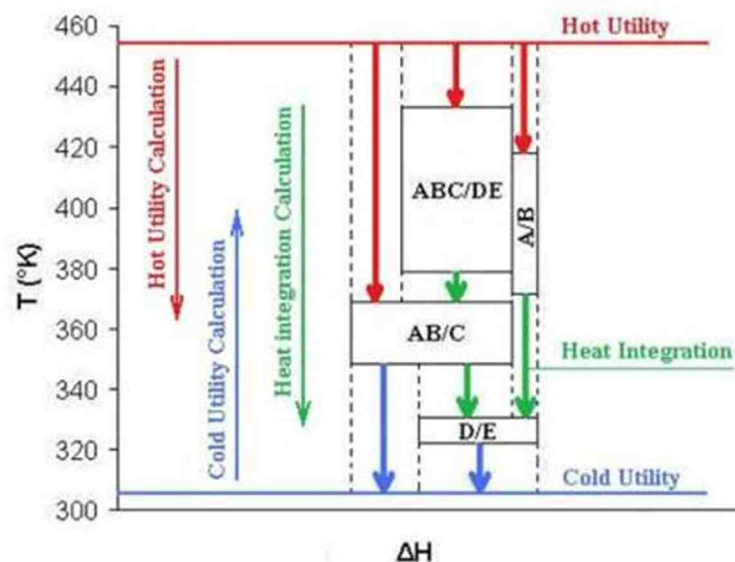


Fig. 6. Calculations directions of proposed algorithm

In the course of searching by an IHS algorithm for solving the optimization problem (8), cost function C must be evaluated effectively for every specific set of values of the variables, $\{s_i\}$, and $\{p\}$. In the present work, for a given set of the codes and pressures, the value of the cost function in equation (8) is estimated by costing of all heat duties. Coding procedure and presented algorithm are the main step for heat duty calculation. Once a code sequence $\{s_i\}$ is given, the corresponding separation sequence and its binary task matrix can be systematically identified. When the recoveries of the pair of key components in each column are specified, the compositions, flow rates and heat duties of all the streams in the flow sheet can be calculated from the first column to the last one. All procedure of this paper for modeling and optimization was coded and optimized by MATLAB.

4. Examples and discussions

To illustrate the accuracy of the proposed algorithm for designing heat integrated distillation system, four typical separation problems are chosen as a case study.

Example 1

Example 1 is a typical five-component separation problem which is optimized by the accurate and time consuming heuristic method of Isla & Cerda (1988), with the total operating cost of 167.40 \$/hr. Input data and cost factors are presented in Table 1. A mixture of five components is to be separated into pure products.

Table 1. Data for examples 1, 2, and 3

Feed components	Mole fraction		
	Case Study 1	Case Study 2	Case Study 3
Propane (A)	0.05	0.15	0.35
i-Butane (B)	0.15	0.20	0.35
n-Butane (C)	0.25	0.30	0.10
i-Pentane (D)	0.20	0.25	0.10
n-Pentane (E)	0.35	0.10	0.10
Feed rate 907.2 kgmol/h			
Utility set	T (K)	Cost (\$/GJ)	
HP-steam	525	4.703	
MP-steam	454	3.313	
LP-steam	382	1.900	
Cooling water	305	0.392	

The problem involves heat integration among four hot streams and four cold streams, several hot utilities and cold ones, depending on the temperature of the reboiler and condenser. In this work, IHSA is used to minimize the objective function of total operating cost. The strings for the optimal result are as follows: Sequence = {3,4,2,1}, Columns pressure = {1.91,0.19,1.36,2.37} MPa. The optimum heat integrated distillation system and the T-ΔH structure are shown in Figures 7 and 8. The network minimizes operating cost to 163.91 \$/hr. The CPU time for solving the objective function was 2.44 seconds by a core 2 due minimizacomputers with 2.5 GHz processor. The Cost - iteration diagram of IHSA for case study 1 is shown in Figure 9

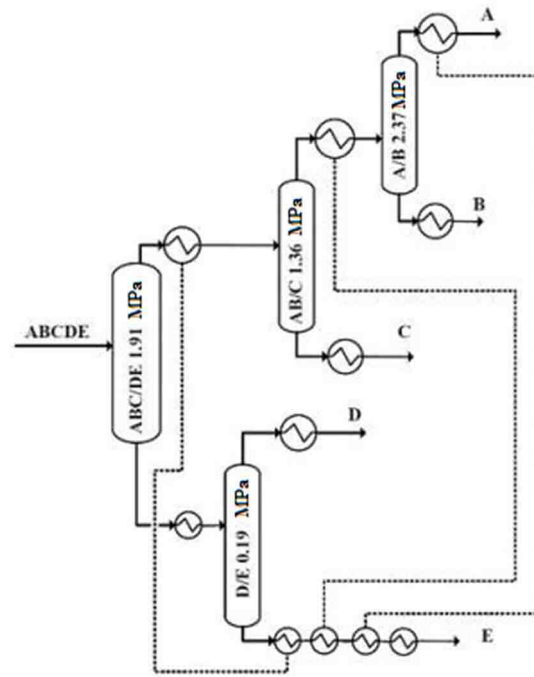


Fig. 7. Optimum heat integrated sequence for case 1

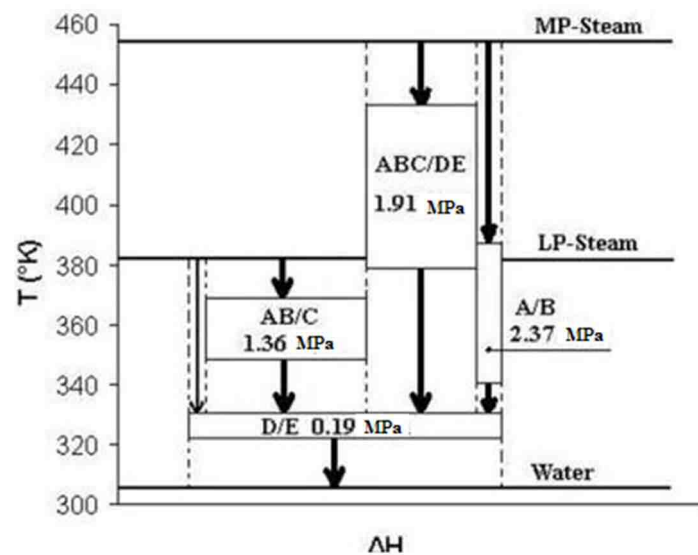


Fig. 8. T- ΔH structure for case 1

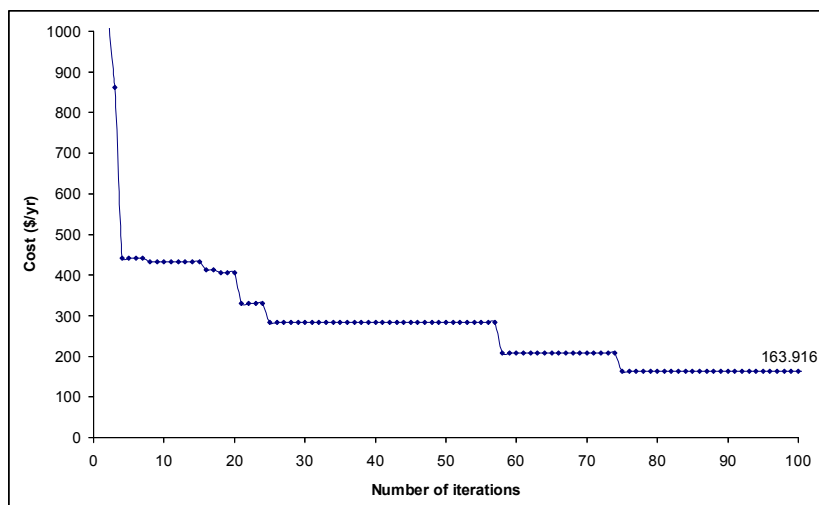


Fig. 9. Cost - iteration diagram of IHSA for case 1

Example 2

Example 2 is the separation of a mixture of five components, as shown in Table 1, which is optimized by the accurate and time consuming heuristic method of Isla & Cerda (1988), with the total operating cost of 154.10 \$/hr.

The optimal heat integrated distillation system and the T- Δ H structure are obtained by the presented algorithm in this work as shown in Figures 10 and 11. The network minimized the operating cost to 149.49 \$/hr. The strings for the optimal result are: Sequence = {1,3,2,4}, Columns pressure = {2.45,1.37,0.61,2.05} MPa. The CPU time for solving the objective function was 33.37 seconds. The Cost - iteration diagram of IHSA for case study 2 is shown in Figure 12.

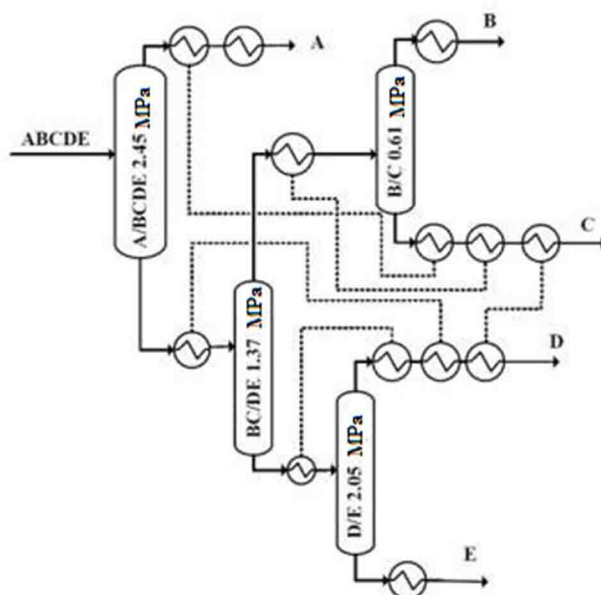


Fig. 10. Optimum heat integrated sequence for case 2

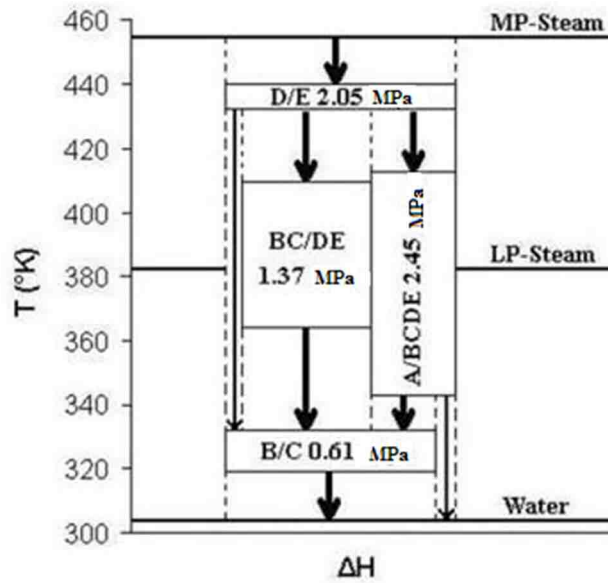


Fig. 11. T- ΔH structure for case 2

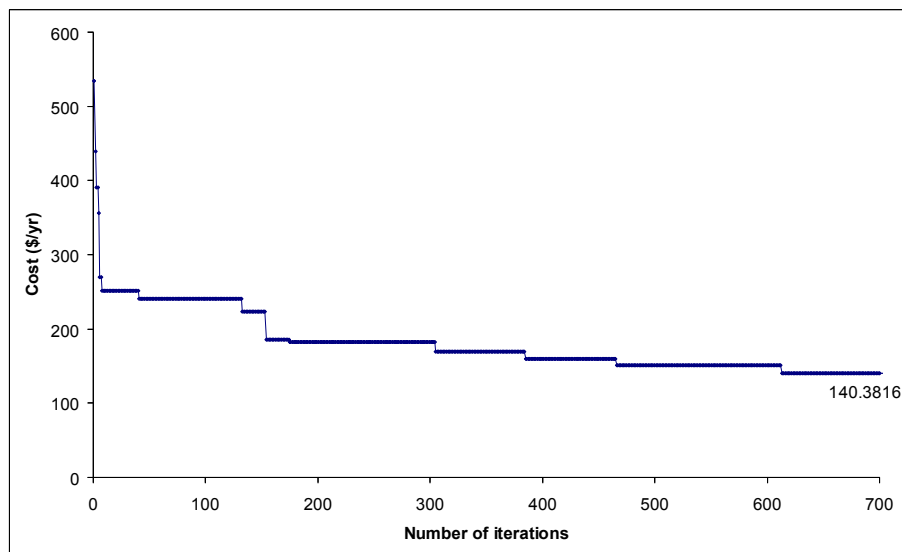


Fig. 12. Cost - iteration diagram of IHSA for case 2

Example 3

Example 3 is a five-component separation problem, as shown in Table 1, which is optimized by Isla & Cerda (1988), with the total operating cost of 138.60 \$/hr.

The optimal heat integrated distillation system and the T- ΔH structure obtained by the presented algorithm in this work are shown in Figures 13 and 14. The network minimized the operating cost to 124.31 \$/hr. The strings for the optimal result are: Sequence = {1,3,2,4},

Columns pressure = {2.89,1.55,0.67,1.98}. The CPU time for solving the objective function was 24.80 seconds. The Cost - iteration diagram of IHSA for case study 3 is shown in Figure 15.

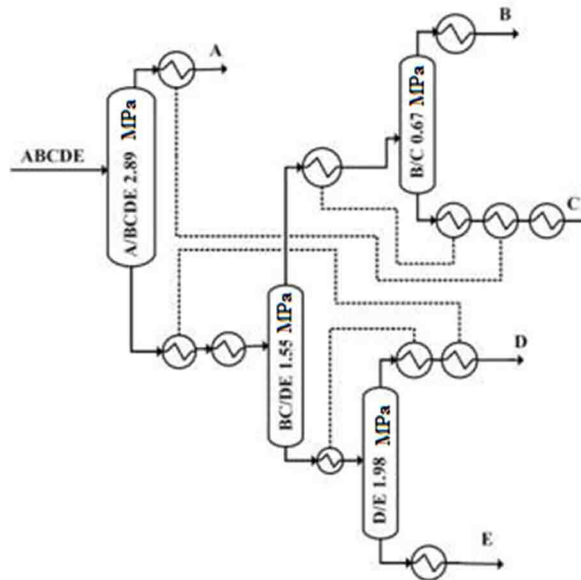


Fig. 13. Optimum heat integrated sequence for case 3

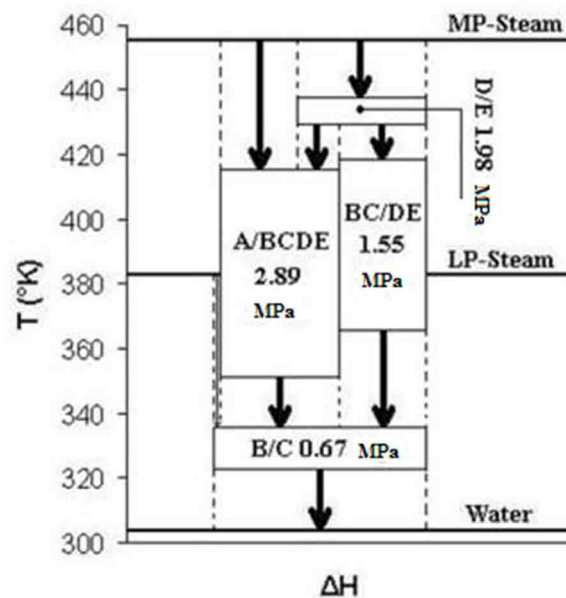


Fig. 14. T-ΔH structure for case 3

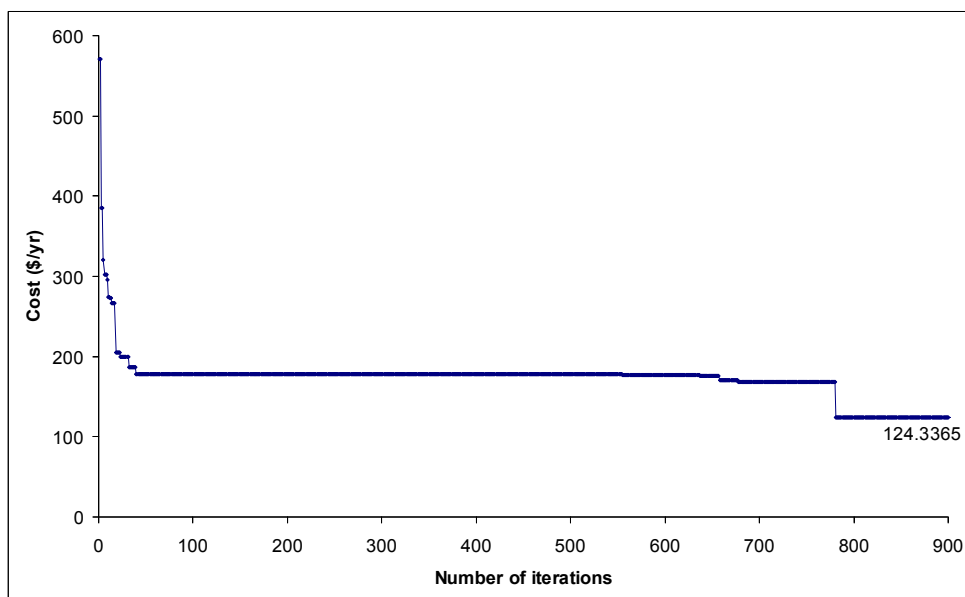


Fig. 15. Cost - iteration diagram of IHSA for case 3

Example 4

Example 4 deals with the separation of a four-component mixture into pure components, which is optimized by Isla & Cerda, (1988), with the total operating cost of 317.83 \$/hr. The mixture contains benzene, toluene, ethylbenzene and o-xylene (Table 2). There are five possible sequences comparing a total number of ten columns for achieving the desired goals.

Table 2. Data for example 4

Feed components	Mole fraction	
Benzene (A)	0.25	
Toluene (B)	0.30	
Ethylebenzene (C)	0.25	
o-xylene (D)	0.20	
Feed rate 453.6 kgmol/h		
Utility set	T (K)	Cost (\$/GJ)
MP-steam	465	3.950
LP-steam	382	2.770
Cooling water	305	0.392

The optimal heat integrated distillation system and the T- Δ H structure obtained by the presented algorithm in this work are shown in Figures 16 and 17. The network minimized the

operating cost to 317.83 \$/hr. The strings for the optimal result are: Sequence = {2,1,3}, Columns pressure = {0.11,0.22,0.23} MPa. The CPU time for solving the objective function was 10.68 seconds. The Cost - iteration diagram of IHSA for case study 5 is shown in Figure 18.

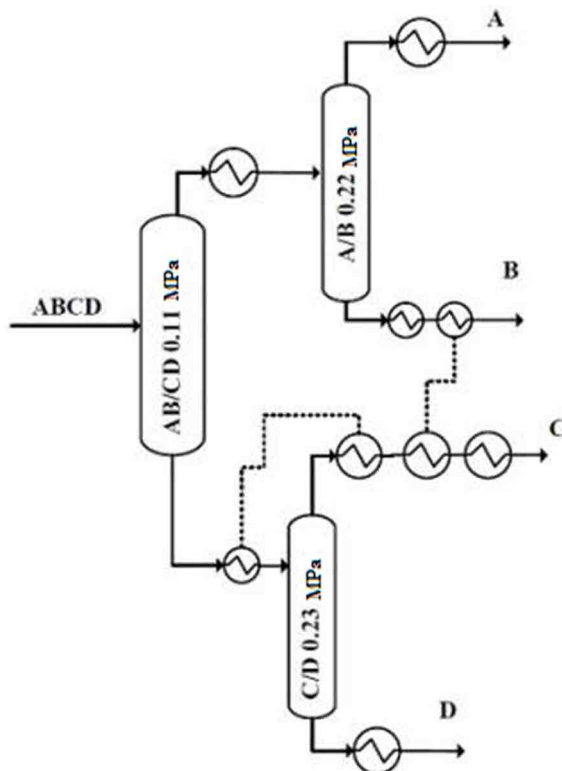


Fig. 16. Optimum heat integrated sequence for case 4

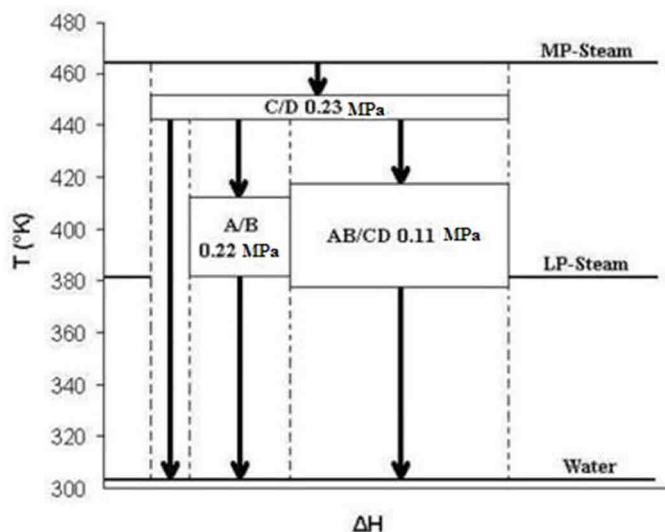


Fig.17. T- ΔH structure for case 4

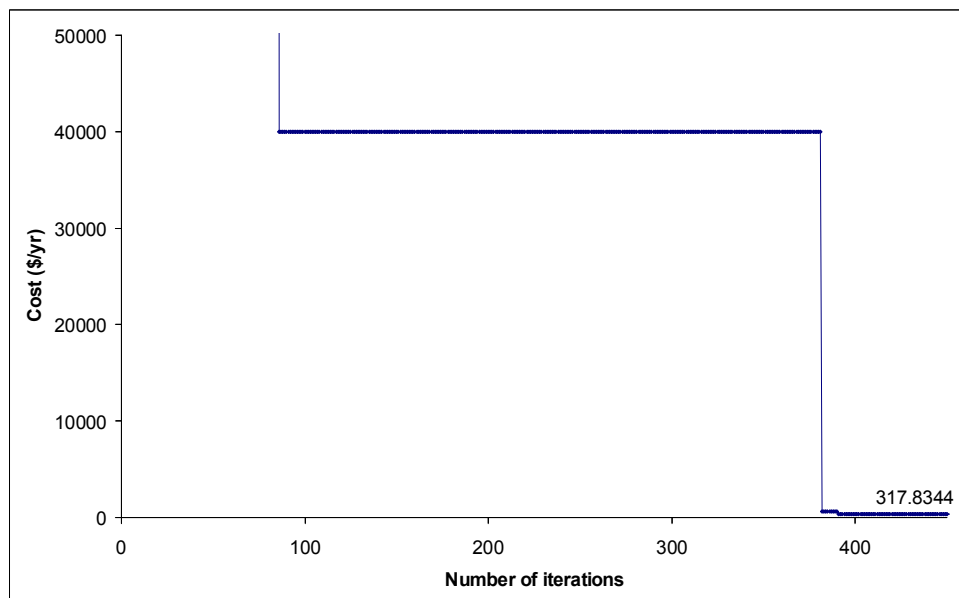


Fig. 18. Cost - iteration diagram of IHSA for case 4

5. Conclusions

The author proposed a new algorithm for the calculation of heat integration among columns and the objective function of total operating cost using the improved harmony search algorithm for minimizing the problem. The simplicity of new algorithm for the calculation of utility consumption and created heat integration among columns decreased the time of computation of cost function. Authors also canceled the discrete parameter of heat exchanger network from the decision parameters of cost model and imported the design of heat exchanger network for internal heat integration into the presented algorithm, which also increased the speed of minimization of objective function by optimization methods. The presented algorithm was examined by four examples. Results of examples showed this method is more precise and faster than presented accurate heuristic method of Isla & Cerda (1988) which takes a lot of time to find the optimum heat integrated sequence.

Nomenclature

Q	Heat flow rate
T	Temperature
C_i	Components
ΔT_{min}	Minimum temperature difference
HMS	Harmony Memory Size
$HMCR$	Harmony Memory Considering Rate
PAR	Pitch Adjusting Rate
NI	Number of solution vector generations
PAR_{min}	Minimum pitch adjusting rate
PAR_{max}	Maximum pitch adjusting rate
bw_{min}	Minimum bandwidth
bw_{max}	maximum bandwidth
gn	Generation Number
$C()$	Cost model for the system
S	A set including all possible flow sheet sequences
P	A set of feasible range of pressures
$\{si\}$	Discrete decisions for the sequences
$\{p\}$	Continuous decision for the individual columns' operating pressures
GA	Genetic Algorithm
$IHSA$	Improved Harmony Search Algorithm

References

- Andreacovich, M.J., & Westerberg, A.W. (1985). An MILP formulation for heat integrated distillation sequence synthesis. *AIChE Journal*, 31, 1461 – 1474.
- Duran, M.A., & Grossmann, I.E. (1986). An Outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36, 307 – 331.
- Giridhar, A., & Agrawal, R. (2010). synthesis of distillation configurations. I: Characteristics of a good search space. *Computers and Chemical Engineering*, 34, 73–95.
- Giridhar, A., & Agrawal, R. (2010). Synthesis of distillation configurations. II: A search formulation for basic configurations. *Computers and Chemical Engineering*, 34, 73–95.
- Geem, Z.W., Kim, J.H., & Loganathan, G.V. (2002). Harmony search optimization: Application to pipe network design. *International Journal of Simulation Modeling*, 22(2), 125–133.
- Grossmann, I.E. (1985). Mixed-integer programming approach for the synthesis of integrated process flowsheets. *Computer and Chemical Engineering Journal*, 9(5), 463 – 475.
- Isla, M.A., & Cerda, J. (1988). A heuristic method for the synthesis of heat-integrated distillation systems. *Chemical Engineering Journal*, 38, 161 – 177.

- Ivakpour, J., & Kasiri, N. (2009). Synthesis of distillation column sequences for nonsharp separations. *Ind. Eng. Chem. Res.*, 48, 8635–8649.
- Jaroslav, J., & Radim P. (1988), Synthesis of heat integrated rectification systems. *Computer and Chemical Engineering Journal*, 12(5), 427 – 432.
- Kocis, G. R., & Grossmann, I.E. (1987). Relaxation strategy for the structural optimization of process flowsheets. *Industrial & Engineering Chemistry Research*, 26, 1869.
- Kravanja, Z., & Grossmann, I.E. (1990). PROSYN – an MINLP process synthesizer. *Computer and Chemical Engineering*, 14, 1363 – 1378.
- Leboreiro, J., & Acevedo, J. (2004). Processes synthesis and design of distillation sequences using modular simulators: A genetic algorithm framework. *Computers and Chemical Engineering*, 28, 1223–1236
- Lee, K.S., & Geem, Z.W. (2005). A new Meta-heuristic algorithm for continues engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 194, 3902–3933.
- Linnhoff, B. et al. (1982). User guide and process integration for efficient use of energy, *Chemical engineering. Oxford: Pergamon Press*.
- Linnhoff, B., Douford H., & Smith R. (1983). Heat integration of distillation column into overall processes. *Chemical Engineering Science*, 38, 1175 – 1188.
- Lucia, A., & McCallum, B. (2010). Energy targeting and minimum energy distillation column sequences. *Computers and Chemical Engineering*, 34, 931–942.
- Liu, Y.A. (1987). *Process synthesis: Some simple and practical developments in recent development in chemical process and plant design*. New York: Wiley.
- Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188, 1567–1579.
- Proios, P., & Efstratios N. Pistikopoulos. (2005). Generalized modular framework for the representation and synthesis of complex distillation column sequences. *Ind. Eng. Chem. Res.*, 44, 4656–4675.
- Qian, Yu., & Lien, K.M. (1994). Application of a fuzzy match inference strategy in synthesis of separation systems. *The Canadian Journal of Chemical Engineering*, 4, 711–721.