Korean J. Chem. Eng., **29**(10), 1298-1309 (2012)
DOI: 10.1007/s11814-012-0007-2

A hybrid optimization strategy for simultaneous synthesis of heat exchanger network

Zhaoyi Huo*, Liang Zhao*, Hongchao Yin*,†, and Jianxiong Ye**

*School of Energy and Power Engineering, Dalian University of Technology, Dalian 116023, China
**School of Mathematical Sciences, Dalian University of Technology, Dalian 116023, China
(*Received 11 July 2011 • accepted 16 January 2012*)

Abstract−The heat exchanger network synthesis problem often leads to large-scale non-convex mixed integer nonlinear programming formulations that contain many discrete and continuous variables, as well as nonlinear objective function or nonlinear constraints. In this paper, a novel method consisting of genetic algorithm and particle swarm optimization algorithm is proposed for simultaneous synthesis problem of heat exchanger networks. The simultaneous synthesis problem is solved in the following two levels: in the upper level, the network structures are generated randomly and reproduced using genetic algorithm; and in the lower level, heat load of units and stream-split heat flows are optimized through particle swarm optimization algorithm. The proposed approach is tested on four benchmark problems, and the obtained solutions are compared with those published in previous literature. The results of this study prove that the presented method is effective in obtaining the approximate optimal network with minimum total annual cost as performance index.

Key words: Heat Exchanger Network Synthesis, Discrete and Continuous Variables, Genetic Algorithm, Particle Swarm Optimization

INTRODUCTION

In the process industry, the heat exchanger network (HEN) is an important daughter system because of its impact on heat energy recovery. HEN synthesis (HENS) is usually a very complicated task that aims to find an economic way for matching hot and cold process streams. Over the past 40 years, two primary methods have been proposed for HENS: sequential and simultaneous approaches [1]. Recently, more and more attention has been paid to simultaneous approaches without any decomposition, which have shown to be superior to sequential approaches in most cases. Simultaneous methods are primarily mixed integer nonlinear programming (MINLP) formulations of the HENS problem. These formulations are nonconvex, and usually have multiple local optima. Furman and Sahinidis analyzed the computational complexity of HENS, and proved that simultaneous HENS belong to NP-hard problems in the strong sense [2]. The objective of this particle is to push the development of efficient global optimization techniques suitable for simultaneous HENS problems.

In general, global optimization techniques may be broadly divided into deterministic and stochastic approaches [3]. Several deterministic methods have been used for solving MINLP problems, such as branch and bound technique [4,5], outer approximation methods [6], branch and reduce algorithm [7] and extended cutting-plane methods [8]. Yee and Grossmann applied a method integrated by penalty function and outer approximation to solving the simultaneous HENS problems with the assumption of isothermal mixing [9]. Zamora and Grossmann presented a hybrid branch and bound/outer-approximation method for simultaneous HENS with no stream splits [10]. The common feature of these MINLP models is that nonlinear terms only exist in the objective function, which makes the MINLP models robust and relatively easy to solve. It should be noted that the assumption of the isothermal mixing or no stream splits will make certain good alternatives in the network configuration to be neglected. Björk and Westerlund presented a global optimization approach for simultaneous synthesis without assumption of isothermal mixing [11]. Removing this assumption imposes some continuous variables and nonlinear constraints on the original model. The heart of the proposed optimization strategy is to convexify the models, and enable the convex models to be solved with standard MINLP-solvers. From the above mentioned, the deterministic methods can be used for the MINLP models of HENS problems. However, the above methods are usually applied to solving the relatively small problems only, among which at most seven process streams are involved.

Compared with deterministic methods, stochastic algorithms, such as simulated annealing approach (SA) [12,13], genetic algorithm (GA) [14-18], different evolution (DE) [19], Tabu search algorithm [20], and randomized algorithm [21-23], are widely used for dealing with medium or large HENS problems. These algorithms do not depend on gradients and work on function evaluation alone. Therefore, stochastic algorithms are effective tools to solve nonconvex optimizations problems. Among existing stochastic algorithms, GA is one of the most well known tools for solving HENS problems. Genetic algorithm is a method to obtain an optimal solution in an adaptive way guided by the equivalent biological evolution mechanisms of reproduction, crossover and mutation. In the search of the GA, the mutation operator works for exploration and the reproduction and crossover operators work for exploitation. Therefore, the GA approach provides a way to possibly obtain the global optimum solution. Moreover, it is observed that the GA is best suited for structural optimization problems with discrete variables [24]. The solution strategies for HENS problems based on stochastic algo-

†To whom correspondence should be addressed.
E-mail: hcyin@dlut.edu.cn

rithms usually have a two-level structure whereby candidate configurations are generated in the upper level and process parameters are optimized in the lower level for each generated structure. In other words, the involved integer and continuous variables are optimized separately by one or more stochastic algorithms. The special advantages and disadvantages of each stochastic algorithm can be considered in hybrid algorithm design. In addition, many hybrid optimization strategies incorporating different deterministic and stochastic algorithms have been developed to enhance the solution quality [25-27].

Recently, researchers in engineering optimization have paid a great deal of attention to the particle swarm optimization (PSO) algorithm, firstly introduced by Kennedy and Eberhart [28]. The algorithm is driven by the social behavior of a bird flock and can be viewed as a population-based stochastic optimization algorithm. The PSO is initialized with a population of random solutions of the fitness function. The individuals are updated by cooperation and competition among the individuals themselves through generations. In PSO, each individual is named as a "particle" which adjusts its own "flying path" according to its flying experience as well as the flying experience of neighboring particles. The PSO is attractive because it has very few parameters to adjust, and easily handle continuous state variables. Numerous studies have successfully improved PSO and applied PSO to various problems [29].

Another important issue is how to handle constraints. Most HENS problems have a large number of equality or inequality constraints from the energy balance, mass balance and logical limitations. The common drawback of stochastic algorithms is unstable performance for solving constrained optimization problems, because the evaluation of the objective function is meaningful only at feasible points. Constraints are mostly handled by using penalty functions, which penalizes infeasible solutions by worsening their fitness values. However, when an optimization problem is highly constrained, the performance of the penalty method is not always satisfactory. It motivates us to develop an appropriate method for handling constraints.

In this paper, splitting streams and non-isothermal mixing are both considered for HENS problem, which is formulated as a nonconvex MINLP model with many inequality and equality constraints. A novel two-level solution approach, making use of GA and PSO, is proposed for solving the proposed model. In the lower level, an improved PSO, which is modified for enhancing convergence to the global optimum, is employed for optimizing process parameters, such as heat exchanger duties and stream flows. In the upper level, the structural variables including exchanger matching and the number of split-stream are updated by GA after enough iterations in process parameters. In addition, to enhance the performance of the proposed algorithms, a new feasible solution strategy together with penalty scheme is proposed for handling constraints.

PROBLEM STATEMENT

The basic HENS problem is generally stated as an integral design of a set of hot process streams needed to be cool down to their desired temperatures, and a set of cold process streams needed to be heat up to their target temperatures. The goal is to obtain a cost-optimal HEN structure defined by stream matches, heat flows of all branches, heat loads and operating temperatures of heat exchangers, areas of heat exchangers and utility consumption. The necessary problem data is composed of inlet temperatures, outlet temperatures, heat flows of process streams, together with cost data of heat exchangers and utilities available.

Yee and Grossmann introduced a stage-wise representation of superstructure not relying on any heuristics [30]. Fig. 1 illustrates a two-stage superstructure involving two hot and two cold process streams. At each stage, hot and cold streams are split to allow the potential match between any pair of hot and cold streams, whereas no split stream is allowed to flow through more than one exchanger. Splitting streams are remixed before entering the next stage without assumption of isothermal mixing. For simplicity in the presentation, utility exchangers are placed on the extreme ends of the whole superstructure. The simultaneous MINLP model is formulated based on the stage-wise representation, in which nonlinear expressions are located both in the objective function and equality constraints.

1. Objective Function

The objective function can be defined as the annual cost including capital and operating costs. The capital cost is usually composed of fixed cost and area cost. Binary variables are introduced to indicate the existence of each potential match, which corresponds to fixed cost for heat exchangers if needed or not. The area of heat

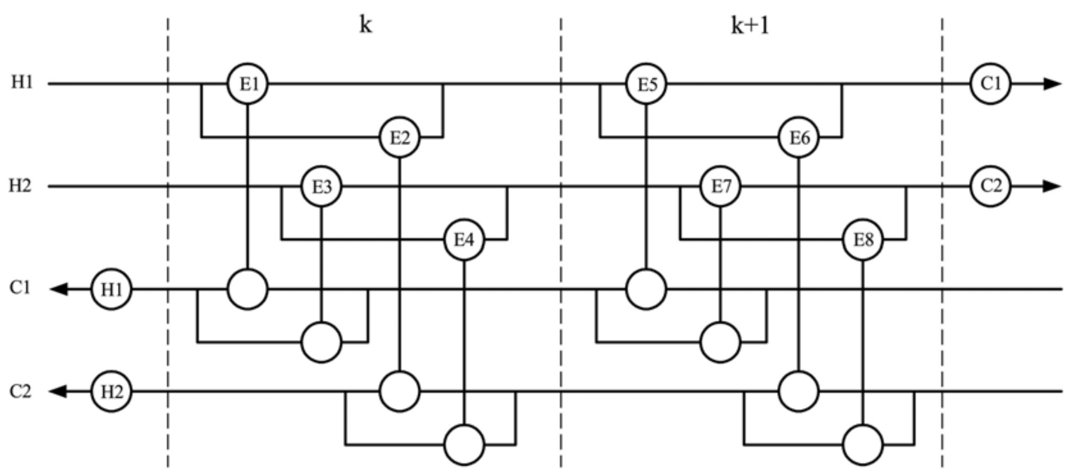

Fig. 1. General stage-wise superstructure.

exchangers is a function of heat load (q), overall heat transfer coefficient (U) and log mean temperature difference (LMTD). The LMTD approximation proposed by Chen is adopted because this formulation tends to be slightly easier to solve under more extreme values of the variables [31]. The operating cost is a simple function of utility loads of the network. The first three terms in Eq. (1) represent capital costs of heat exchangers, heaters and coolers, and the next two terms represent operating costs of hot and cold utility available.

$$\mathrm{Min}\left\{\sum_{i=1}^{HN}\sum_{j=1}^{CN}\sum_{k=1}^{KN}[C_{ef}Be_{i,j,k}+C_{ea}(A^{e}_{i,j,k})^{He}]+\sum_{i=1}^{HN}[C_{cf}Bcu_i+C_{ca}(A^{cu}_i)^{Hcu}] +\sum_{j=1}^{CN}[C_{hf}Bhu_j+C_{ha}(A^{hu}_j)^{Hhu}]+\sum_{i=1}^{HN}C_{cu}q^{cu}_i+\sum_{j=1}^{CN}C_{hu}q^{hu}_j\right\} \quad (1)$$

$$A^{e}_{i,j,k}=\frac{q_{i,j,k}}{U_{i,j}LMTD_{i,j,k}}$$
$$(i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \quad (2)$$

$$LMTD_{i,j,k}\approx\left[(t^{h}_{i,k,out}-t^{c}_{j,k,out})(t^{h}_{i,k+1,out}-t^{c}_{j,k+1,in})\left(\frac{t^{h}_{i,k,out}+t^{h}_{i,k+1,out}}{2}-\frac{t^{c}_{j,k,out}-t^{h}_{i,k+1,out}}{2}\right)\right]^{1/3}$$
$$(i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \quad (3)$$

where C_{ef}, C_{cf}, C_{hf} are separately fixed charge for heat exchanger, heater and cooler; C_{ea}, C_{ca}, C_{ha} are separately area cost efficient for heat exchanger, heater and cooler; A^{e}, A^{cu}, A^{hu} are area of heat exchanger, heater and cooler; Be, Bcu, Bhu are binary variables representing the existence of heat exchanger, heater and cooler; t^{h} is temperature of hot process stream, t^{c} is temperature of cold process stream; i represents hot process stream, j represents cold streams, k represents stage in the superstructure; HN is the number of hot process streams, CN is the number of cold process streams, and KN is the number of stages.

2. Constraints Formulation

Eqs. (4)-(19) are equality constraints coming from energy balance and mass balance, and Eqs. (20)-(33) are inequality constraints for specifying feasible and logical operation.

2-1. Overall Heat Balance for Each Stream

An overall heat balance makes sure that each process stream receives enough amount of heat or cold for reaching their desired temperatures. The required amount of heat or cold is supplied from process or utility streams.

$$(T^{h}_{i,in}-T^{h}_{i,out})W^{h}_i=\sum_{k=1}^{KN}\sum_{j=1}^{CN}q_{i,j,k}+q^{cu}_i \quad (i=1, 2, \ldots, HN) \quad (4)$$

$$(T^{c}_{j,out}-T^{c}_{j,in})W^{c}_j=\sum_{k=1}^{KN}\sum_{i=1}^{HN}q_{i,j,k}+q^{hu}_j \quad (j=1, 2, \ldots, CN) \quad (5)$$

where T^{h}, T^{c} are separately given temperatures of hot and cold process streams; in represents inlet, out represents outlet; W^{h}, W^{c} are separately heat flows of hot and cold process streams.

2-2. Heat Balance at Each Stage

Each process stream is split at the inlet side of each stage, and the splitting streams are remixed before entering the next stage. The total amount of heat exchanged equals the sum of the heat loads of the exchangers existing on the splits.

$$(t^{h}_{i,k,in}-t^{h}_{i,k,out})W^{h}_i=\sum_{j=1}^{CN}q_{i,j,k} \quad (i=1, 2, \ldots, HN; k=1, 2, \ldots, KN) \quad (6)$$

$$(t^{c}_{j,k,out}-t^{c}_{j,k,in})W^{c}_j=\sum_{i=1}^{HN}q_{i,j,k} \quad (j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \quad (7)$$

2-3. Heat Balance for Utility Load

The amount of utilities required can be determined by the outlet temperatures of the superstructure and the target temperatures of process streams.

$$(t^{h}_{i,KN,out}-T^{h}_{i,out})W^{h}_i=q^{cu}_i \quad (i=1, 2, \ldots, HN) \quad (8)$$

$$(T^{c}_{j,out}-t^{c}_{j,1,out})W^{c}_j=q^{hu}_j \quad (j=1, 2, \ldots, CN) \quad (9)$$

2-4. Heat Balance for Each Exchanger

When a stream split exists without the assumption of isothermal mixing, a nonlinear heat balance around each exchanger should be satisfied.

$$(tp^{h}_{i,j,k,in}-tp^{h}_{i,j,k,out})Wp^{h}_{i,j,k}-q_{i,j,k}$$
$$(i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \quad (10)$$

$$(tp^{c}_{i,j,k,out}-tp^{c}_{i,j,k,in})Wp^{c}_{i,j,k}-q_{i,j,k}$$
$$(i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \quad (11)$$

where tp represents temperature of split stream, Wp represents heat flows of slit stream.

2-5. Mass and Energy Balance for Each Splitter/Mixer

When a stream split exists without the assumption of isothermal mixing, the following equalities from heat balance should be satisfied.

$$W^{h}_i\cdot t^{h}_{i,k,out}=\sum_{j=1}^{CN}Wp^{h}_{i,j,k}\cdot tp^{h}_{i,j,k,out} \quad (i=1, 2, \ldots, HN; k=1, 2, \ldots, KN) \quad (12)$$

$$W^{c}_j\cdot t^{c}_{j,k,out}=\sum_{i=1}^{HN}Wp^{c}_{i,j,k}\cdot tp^{c}_{i,j,k,out} \quad (j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \quad (13)$$

$$W^{h}_i=\sum_{j=1}^{CN}Wp^{h}_{i,j,k} \quad (i=1, 2, \ldots, HN; k=1, 2, \ldots, KN) \quad (14)$$

$$W^{c}_j=\sum_{i=1}^{HN}Wp^{c}_{i,j,k} \quad (j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \quad (15)$$

Note that Eqs. (12) and (13) are also nonlinear.

2-6. Assignment of Superstructure Temperatures

Supplying temperatures of the process streams are assigned as the inlet temperatures of the superstructure. For hot streams, the superstructure inlet corresponds to temperature location k=1, and the inlet temperature of each stage corresponds to the outlet temperature of the left adjacent stage, while for cold streams, the inlet corresponds to location k=KN, and the inlet temperature of each stage corresponds to the outlet temperature of the right adjacent stage.

$$tp^{h}_{i,j,1,in}=t^{h}_{i,1,in}=T^{h}_{i,in} \quad (i=1, 2, \ldots, HN; j=1, 2, \ldots, CN) \quad (16)$$

$$tp^{c}_{i,j,KN,in}=t^{c}_{j,KN,in}=T^{c}_{i,in} \quad (i=1, 2, \ldots, HN; j=1, 2, \ldots, CN) \quad (17)$$

$$tp^{h}_{i,j,k+1,in}=t^{h}_{i,k+1,in}=t^{h}_{i,k,out}$$
$$(i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN-1) \quad (18)$$

$$tp^{c}_{i,j,k-1,in}=t^{c}_{j,k-1,in}=t^{c}_{j,k,out}$$
$$(i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=2, \ldots, KN) \quad (19)$$

2-7. Minimum Approach Temperature Constraints

Eqs. (20)-(25) define that the temperature difference on each side of heat exchangers should not be lower than minimum fixed temperature approach. This is required to obtain appropriate heat transfer area of heat exchangers.

$$tp^{h}_{i,j,k,in} - tp^{c}_{i,j,k,out} \geq \mathrm{EMAT} \quad (i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \tag{20}$$

$$tp^{h}_{i,j,k,out} - tp^{c}_{i,j,k,in} \geq \mathrm{EMAT} \quad (i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \tag{21}$$

$$t^{h}_{i,KN,out} - \mathrm{T\_cool\_out} \geq \mathrm{EMAT} \quad (i=1, 2, \ldots, HN) \tag{22}$$

$$T^{h}_{i,out} - \mathrm{T\_cool\_in} \geq \mathrm{EMAT} \quad (i=1, 2, \ldots, HN) \tag{23}$$

$$\mathrm{T\_heat\_out} - t^{c}_{j,1,out} \geq \mathrm{EMAT} \quad (j=1, 2, \ldots, CN) \tag{24}$$

$$\mathrm{T\_heat\_in} - t^{c}_{j,out} \geq \mathrm{EMAT} \quad (j=1, 2, \ldots, CN) \tag{25}$$

where T_heat_in, T_heat_out are inlet and outlet temperatures of hot utility; T_cool_in, T_cool_out are inlet and outlet temperatures of cold utility; EMAT is minimum fixed temperature approach for each match.

2-8. Feasibility of Temperatures

The following constraints arise from the fact that the temperature of hot (resp. cold) streams decreases (resp. increases) in a monotonic way, and the outlet temperature of the superstructure should not be lower (resp. higher) than the target temperature of hot (resp. cold) streams.

$$t^{h}_{i,k,in} \geq t^{h}_{i,k,out} \quad (i=1, 2, \ldots, HN; k=1, 2, \ldots, KN) \tag{26}$$

$$t^{c}_{j,k,in} \geq t^{c}_{j,k,out} \quad (j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \tag{27}$$

$$t^{h}_{i,KN,out} \leq T^{h}_{i,out} \quad (i=1, 2, \ldots, HN) \tag{28}$$

$$t^{c}_{j,1,out} \geq T^{c}_{j,out} \quad (j=1, 2, \ldots, CN) \tag{29}$$

2-9. Logical Constraints

The binary variables are introduced to determine the existence of potential match in each stage. The match is present only when the corresponding binary variable equals to 1. Additionally, the following logical constraints should be satisfied.

$$q_{i,j,k} \leq Q_{max} Be_{i,j,k} \quad (i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \tag{30}$$

$$q^{cu}_{i} \leq Q_{max} Bcu_{i} \quad (i=1, 2, \ldots, HN) \tag{31}$$

$$q^{hu}_{j} \leq Q_{max} Bhu_{j} \quad (j=1, 2, \ldots, CN) \tag{32}$$

$$Be_{i,j,k}, Bcu_{i}, Bhu_{j} = 0, 1 \quad (i=1, 2, \ldots, HN; j=1, 2, \ldots, CN; k=1, 2, \ldots, KN) \tag{33}$$

where Q_{max} is an upper bound for heat exchanger.

HYBRID METHODOLOGY

The above model is a non-convex MINLP problem with a great number of equality and inequality constraints. The combinatorial nature of mixed-integer programming and non-convex nonlinear programming significantly increases the complexity of computation. The stochastic methods are elected for the proposed model because of their special advantage for solving complex MINLP problems. To avoid trapping into a local optimum prematurely, a two-level method is proposed for optimizing integer and continuous variables separately, and giving adequate evolution for continuous variables, the general procedure of which is presented in Fig. 2. In the upper level of the present method, GA is used for structural optimization, since it is more suitable for the non-continuous optimization problem than other stochastic algorithms. The information of structures is sent to the lower level, in which, the process parameters such as stream-split flows and heat loads of heat exchangers are optimized by an improved PSO algorithm. This algorithm is a powerful tool for solving continuous and unconstrained optimization pro- blems.

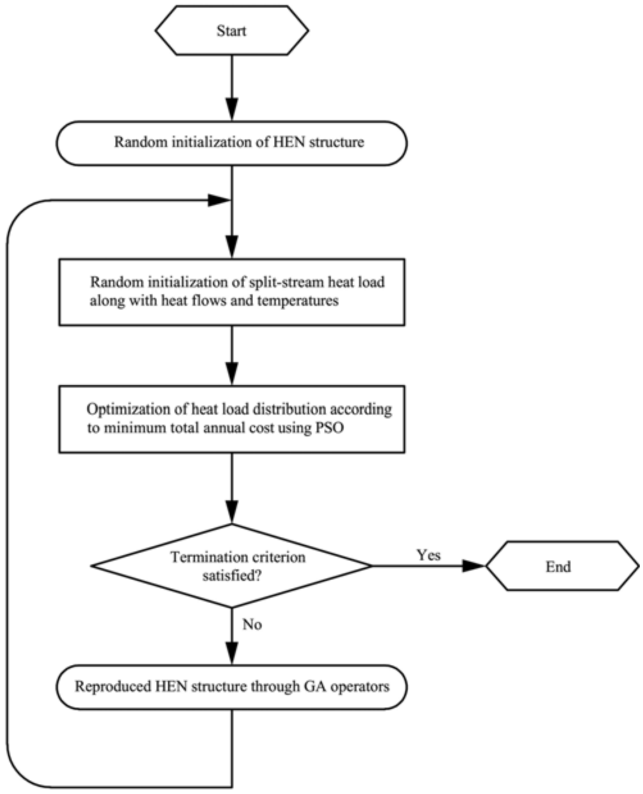

Fig. 2. General structure of the proposed two-level method.

1. Upper-level Network Structural Optimization

In this section, we take each candidate network structure as an individual, the genes of which are potential matches between process streams. In detail, for a HEN superstructure with HN hot streams

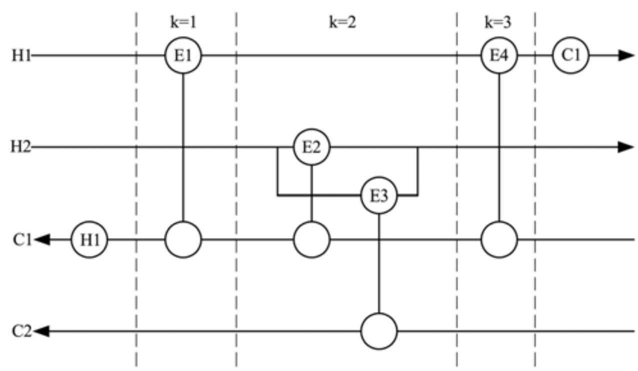

Fig. 3. A sample network structure with four process streams.

Hot stream \ Cold stream	k=1: 1	k=1: 2	k=2: 1	k=2: 2	k=3: 1	k=3: 2
1	1	0	0	0	1	0
2	0	0	1	1	0	0

Fig. 4. An incidence matrix representing network structure.

and CN cold streams, we use an HN×CN matrix, denoted by Be_k, to represent the potential matches between the hot and cold streams at stage k, the elements of which, $Be_{i,j,k}$, takes value 1 if the hot steam i is matched with the cold steam j, and 0 otherwise. As shown in Fig. 3, the sample HEN structure has two hot streams, two cold streams, and three stages. The corresponding matrices are shown as Fig. 4. However, for being manipulated conveniently by GA operators, the structures can also be represented in chromosomal-like fashion. The length of the chromosomes is equal to HN×CN×KN.

1-1. Initialization

An initial population of incidence matrices of binary variables is generated randomly to determine the possible matches between streams at all stages.

$$Be_{i,j,k}=\begin{cases}1 \text{ if rand} <= 0.5\\ 0 \text{ otherwise}\end{cases} \tag{34}$$

Here, rand is a random number uniformly generated from [0, 1].

1-2. Fitness Evaluation

Once the population is generated, the next step is to evaluate the solutions using fitness function. For HENS problem, the fitness of an individual can be defined as a function of the annual cost as follows:

$$\text{fitness}=\frac{1}{1+C+C_{min}},\quad C>0,\ \ C+C_{min}\geq 0. \tag{35}$$

Here, C_{min} is a conservative estimation of objective boundary, which is the minimum observed value of the objective function at the end of each iteration; C is the total annual cost of the individual, which is obtained in the lower level.

1-3. Selection

Selection is the first genetic operator in GA, which aims to determine better chromosomes as parents for reproduction. Roulette-wheel is the simplest proportionate selection scheme. According to this scheme, the individuals of the population are assumed as slots of the Roulette-wheel. The probability P_i of each slot being selected is proportional to its relative fitness, which can be calculated by Eq. (36).

$$P_i=\frac{\text{fitness}_i}{\sum_{i=1}^{n}\text{fitness}_i}. \tag{36}$$

Here n is the size of the population. The selected individuals are directly sent into the child population in a given selection probability of P_s, or to reproduce child individuals through crossover operator in the probability of $(1-P_s)$.

1-4. Crossover

Crossover is the second genetic operator, which takes two parent individuals in a crossover probability p_c and cuts their chromosome strings at some randomly chosen position. Then the corresponding segments swap to produce two child individuals. In this paper, single one point crossover operator, which is the simplest discrete recombination scheme, is adopted. A single random cut is made, producing two head sections and two tail sections. The two tail sections are then exchanged to obtain offsprings.

1-5. Mutation

The final evolutionary operator is mutation, which obtains new individuals by changing one or more genes of each chromosome. In this work, the scheme of uniform mutation, that is, all genes have an equal chance p_m for mutation, is adopted. The mutation probability p_m is usually recommended to be a small number for enhancing local search (e.g. p_m=0.005).

$$Be_{i,j,k}=1-Be_{i,j,k},\ \ \text{if rand}\leq p_m. \tag{37}$$

2. Lower-level Process Parametric Optimization

In the lower level, the process parameter optimization is a nonlinear programming (NLP) problem with equality and inequality constraints. An improved PSO algorithm (IPSO) is used for the NLP problem. To make full use of a favorable PSO algorithm, it is necessary to find an effective method for handling the constraints. Note that the nonlinear terms in the constraints are all bilinear ones. Naturally, we can decompose the continuous variables into independent and dependent variables, and the dependent ones can be expressed explicitly through equality constraints. For this NLP problem, stream splitting flows, the outlet temperatures of the cold process streams and the inlet temperatures of the hot process streams are identified to be independent variables, and the other variables can be solved directly through the equality constraints as follows:

The independent variables are initialized according to Eqs. (38)-(49).

2-1. Stream Splitting Flows

At the stage k for the hot stream i, the split heat capacity flows are initialized by

$$Wp^h_{i,j,k}=\begin{cases}\text{rand}\cdot W^h_i & \text{if } Be_{i,j,k}=1\\ 0 & \text{if } Be_{i,j,k}=0\end{cases} \tag{38}$$

At the stage k for the cold stream j, the split heat capacity flows are initialized by

$$Wp^c_{i,j,k}=\begin{cases}\text{rand}\cdot W^c_i & \text{if } Be_{i,j,k}=1\\ 0 & \text{if } Be_{i,j,k}=0\end{cases} \tag{39}$$

2-2. Outlet Temperatures of the Cold Split Streams

At the stage k, the outlet temperature of the cold stream split is initialized by

$$tp^c_{i,j,k,out}=\begin{cases}(T^c_{j,in}+\text{rand}(tp^h_{i,j,k,in}-\text{EMAT}-T^c_{j,in})) & \text{if } Be_{i,j,k}=1\\ 0 & \text{if } Be_{i,j,k}=0\end{cases} \tag{40}$$

At the stage k, the inlet temperature of the hot stream split is initialized by

$$tp^h_{i,j,k,in}=\begin{cases}t^h_{i,k,in} & \text{if } Be_{i,j,k}=1\\ 0 & \text{if } Be_{i,j,k}=0\end{cases} \tag{41}$$

Once the independent variables are initialized, the other variables can be directly calculated according to Eqs. (42)-(49).

2-3. Heat Load of Heat Exchangers

The maximum heat load between the hot stream i and the cold stream j at the stage k is determined by

$$\text{if } Wp^h_{i,j,k} \geq Wp^c_{i,j,k}$$

$$q\,max_{i,j,k} = \min\begin{pmatrix} Wp^h_{i,j,k}(tp^h_{i,j,k,in} - T^h_{i,out}), \\ Wp^c_{i,j,k}(tp^c_{i,j,k,out} - T^c_{j,in}) \end{pmatrix}$$

else

$$q\,max_{i,j,k} = \min\begin{pmatrix} Wp^h_{i,j,k}(tp^h_{i,j,k,in} - T^h_{i,out}), \\ Wp^c_{i,j,k}(tp^c_{i,j,k,out} - T^c_{j,in}), \\ \dfrac{tp^h_{i,j,k,in} - EMAT - tp^c_{i,j,k,out}}{\dfrac{1}{Wp^h_{i,j,k}} - \dfrac{1}{Wp^c_{i,j,k}}} \end{pmatrix} \tag{42}$$

end

Heat load of the exchangers can be randomly generated by

$$q_{i,j,k} = rand \cdot q\,max_{i,j,k} \tag{43}$$

2-4. Temperatures of Exchangers

The inlet and outlet temperatures of the heat exchangers can be calculated through the equality constraints of energy balances for each exchanger.

$$tp^h_{i,j,k,out} = tp^h_{i,j,k,in} + q_{i,j,k}/Wp^h_{i,j,k} \tag{44}$$

$$tp^c_{i,j,k,in} = tp^c_{i,j,k,out} - q_{i,j,k}/Wp^c_{i,j,k} \tag{45}$$

2-5. Temperatures of Stages

The inlet and outlet temperatures of stages can be calculated through the equality constraints of energy balances for each mixer.

$$t^h_{i,k,out} = \frac{\sum_{j=1}^{CN} Wp^h_{i,j,k} \cdot tp^h_{i,j,k,out}}{W^h_i}, \quad t^h_{i,k+1,in} = t^h_{i,k,out} \tag{46}$$

$$t^c_{i,k,in} = \min(tp^c_{i,1,k,in}, tp^c_{i,2,k,in}, \ldots, tp^c_{i,CN,k,in}), \ \text{ if } tp^c_{i,j,k,in} > 0 \tag{47}$$

2-6. Heat Loads of Utilities

The utility requirements can be calculated through the equality constraints of energy balances constraints of utility load

$$q^{cu}_i = W^h_i(t^h_{i,kN} - T^h_{i,out}) \tag{48}$$

$$q^{hu}_j = W^c_j(t^c_{j,1} - T^c_{j,out}) \tag{49}$$

So far, the equality constraints are eliminated. Then, penalty function methods can be implemented efficiently to solve the NLP problem with inequality constraints by PSO.

In the PSO, each individual is called a particle and a group of particles forms a swarm. Assume that a swarm consists of m particles, each of which is a D-dimensional vector denoted by $X_{id}=(x_{i1}, x_{i2}, \ldots, x_{iD})$ (i=1, 2, …, m). The particles' fitness could be calculated through position vectors using objective function. $Pbest_{id}=(pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD})$ represents the best position of the ith particle and $Gbest_i=(gbest_1, gbest_2, \ldots, gbest_D)$ indicates the best position in the swarm. Let v_{id} be the velocity vector for the particle i (i=1, 2, …, m), then the update of each particle and its velocity can be carried out by the following equations:

$$v_{id}(t+1) = w(t+1)v_{id}(t) + c_1 r_1(Pbest_{id}(t) - x_{id}(t)) + c_2 r_2(Gbest_i - x_{id}(t)) \tag{50}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{51}$$

where c_1 and c_2 are two positive constant, called the cognitive and social acceleration coefficients, respectively; r_1 and r_2 represent uniform random numbers between 0 and 1; w is an inertia weight controlling the influence of previous velocity on the new velocity, which is defined by a linear decreasing function of the iteration t as follows [32]:

$$w(t+1) = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times (t+1) \tag{52}$$

Here, w_{max} and w_{min} are, respectively, the maximum and minimum inertia; t is current iteration time; $iter_{max}$ is the maximum number of iterations. Particles are renewed in each iterative operation until t exceeds the specified $iter_{max}$.

Ratnaweera et al. introduced time-varying acceleration coefficients as follows [33]:

$$c_1(t) = c_{1s} + \frac{c_{1e} - c_{1s}}{iter_{max}} \times (t) \tag{53}$$

$$c_2(t) = c_{2s} + \frac{c_{2e} - c_{2s}}{iter_{max}} \times (t) \tag{54}$$

where c_{1s}, c_{1e}, c_{2s} and c_{2e} are constants. This slight modification of the standard PSO model can avoid premature convergence in the early stages of the search and enhance convergence to the global optimum solution during the later stages of the search.

During the iterative procedure of process parameter optimization, the objective function of minimum annual cost is evaluated to determine the current best position of each particle and the current global best optimization in the swarm. Then the independent variables are updated according to Eqs. (50)-(54), and the dependent variables are calculated directly by the dependent ones through equality constraints, until the criterion of maximum iteration is met.

CASE STUDIES

In this section, four benchmark problems reported in published literature are presented to test performance of the proposed method, which is implemented in a Matlab program written for Mathematical Matlab (version 7.7.0) running on windows XP system. The personal computer has Intel core 2 Duo E6550 2.33GGz equipped with 2 GB RAM.

1. Case 1

The first example is one of the simplest HEH problems, which involves two hot and two cold streams with steam and water as utilities. The data for this problem are given in Table 1, including cost information for both exchangers and utilities.

The numbers of stages and branches determine the possible matches between hot/cold streams, which can be increased to expand searching space for finding global optimum. However, the model size and computation time will increase rapidly as well. Yee et al. advised that the number of stages is seldom chosen to be larger than the number of hot/cold streams, since an optimal design usually does not require a large number of heat exchangers [8]. But for this problem with two hot and two cold streams, the number of stages is initially set to 3 and the maximum number of branches is set to 2 at each stage, which is proved to obtain better solutions effectively. The

Table 1. Problem data for case 1

Stream	T_{in} (K)	T_{out} (K)	W (kW/K)
H1	533	433	3.0
H2	523	403	1.5
C1	393	508	2.0
C2	453	513	4.0
Steam	553	552	-
Water	303	353	-

U=0.2 kW/m^2·K for all matches; Exchanger cost ($)=300A$^{0.5}$ for all matches; Chu=110 $/kW·year; Ccu=12.2 $/kW·year

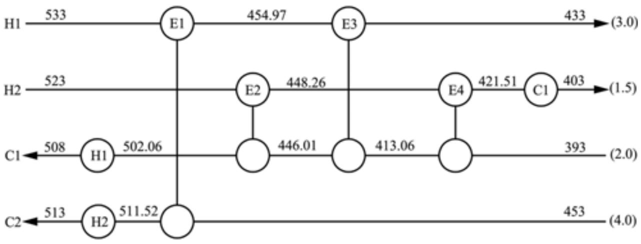

Fig. 5. Optimum HEN structure for case 1.

Table 2. Optimal results for case 1

Method	No. of units	q^{hu} (kW)	q^{cu} (kW)	Annual cost ($)
Ahmad [34]	7	50	60	12870
Nielsen et al. [35]	8	36	45	12306
Khorasany and Fesanghary [26]	7	18.1	28.1	11895
This work	7	17.8	27.8	11632

final network is obtained with annual cost of $11632, which is shown in Fig. 5. Note that no branch is located on the streams in the optimum structure. The reported results with different methods are presented in Table 2 [26,34,35]. The design obtained in this work has the lowest annual cost, the lowest utilities consumption and the lowest number of units.

Table 3. Problem data for case 2

Stream	T_{in} (K)	T_{out} (K)	W (kW/K)	h (kW/m^2·K)
H1	600	313	100	0.50
H2	493	433	160	0.40
H3	493	333	60	0.14
H4	433	318	400	0.30
C1	373	573	100	0.35
C2	308	437	70	0.70
C3	358	411	350	0.50
C4	333	443	60	0.14
C5	413	573	200	0.60
Hot oil	603	523	-	0.50
Water	288	303	-	0.50

Lifetime of plant=5 years; Rate of interest=0%; Annual cost ($)= 10,000+350A for all exchangers; Chu=60 $/kW·year; Ccu=6 $/kW·year

2. Case 2

This case is one popular medium-scale HENS problem, firstly studied by Linhoff and Ahmad [36], which involves four hot and five cold streams with hot oil and cold water as utilities. The supplying data for the problem is presented in Table 3.

Compared with the first example, the solution space of this problem is significantly larger because of the increased number of process streams. Adequate numbers of stages and branches for each stream are necessary to obtain a near optimal solution. However, considering excessive potential matches of hot/cold streams might not obtain better solution because it implies more possible heat ex-

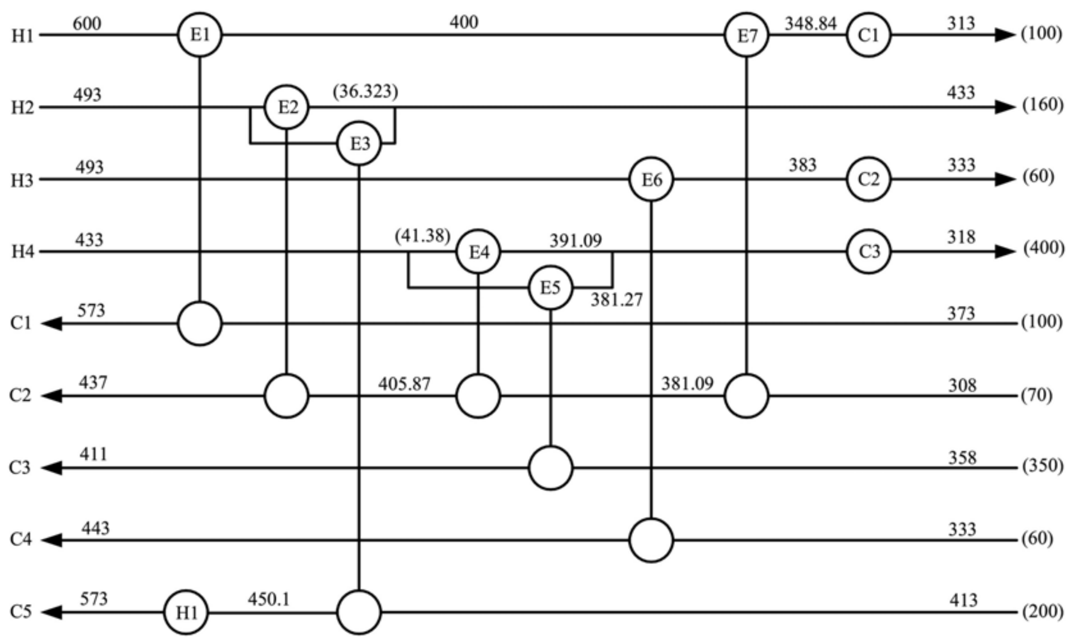

Fig. 6. Optimum HEN structure for case 2.

Table 4. Optimal results for case 2

Method	No. of units	Total area (m^2)	q^{hu} (MW)	q^{cu} (MW)	Annual cost (M$)
Linhoff and Ahmad [36]	13	17400	25.31	33.03	2.960
Yerramsetty and Murty [19]	15	16536	25.88	33.60	2.942
Lewin [15]	12	17050	25.09	32.81	2.936
X. Luo et al. [25]	14	-	23.62	31.34	2.922
Pettersson [37]	17	16880	24.27	31.99	2.905
This work	11	12546	24.58	32.30	2.9256

changers in the network, which would never appear in the optimized solution. For this problem, the maximum number of stages is initially set to three, and the maximum number of branches for each stream is assigned to two. The final network structure with an annual cost of 2925634 is obtained, shown in Fig. 6. This structure has three stages and split-stream exists in stage 1 and stage 2. As indicated in Table 4, the optimal annual cost in this paper is less than that in the literature but one introduced by Petterson [37]. The optimal structure obtained by Petterson contains the configuration that a split stream goes through two exchangers in series, which is not considered in our superstructure.

3. Case 3

This case, originally proposed by Linnhoff and Flower [38], contains five hot and five cold streams. It is also a popular problem studied by many researchers. The input data is summarized in Table 5. The maximum number of stages is set to three and the maximum number of branches in each stage is set to two.

The final structure is shown in Fig. 7. This structure has three stages and split-stream only exits in stage 1. The total annual cost for the obtained network is $43431, which is only slightly higher

Table 5. Problem data for case 3

Stream	T_{in} (K)	T_{out} (K)	W (kW/K)
H1	433	366	8.79
H2	522	411	10.55
H3	544	422	12.56
H4	500	339	14.77
H5	472	339	17.73
C1	355	450	17.28
C2	366	478	13.90
C3	311	494	8.44
C4	333	433	7.62
C5	389	495	6.08
Steam	509	509	-
Cold water	311	355	-

U=0.852 kW/(m^2K) for all matches except for those involving steam; U=1.136 kW/m^2·K for all matches involving steam; Annual cost ($)= 145.63$A^{0.6}$ for all exchangers; Chu=37.64 $/kW·year; Ccu=18.12 $/kW·year

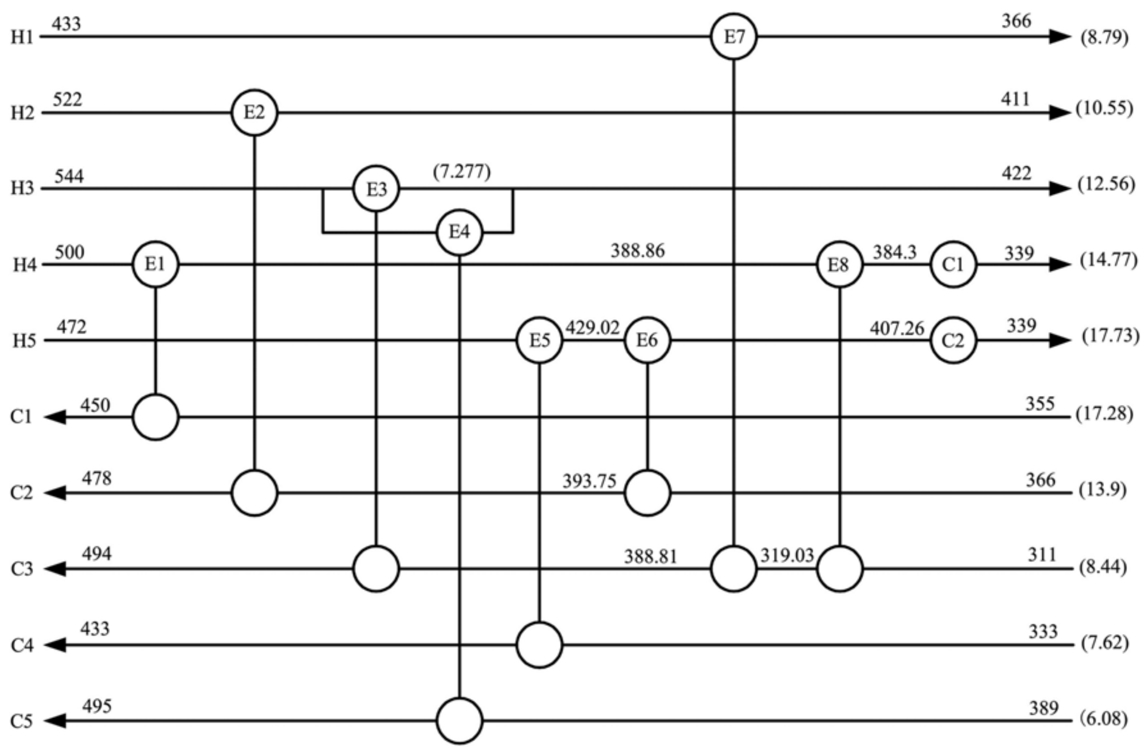

Fig. 7. Optimum HEN structure for case 3.

than the result got by Lin and Miller [20]. The comparison between the result obtained from this paper and those obtained by other researchers is presented in Table 6. The utilities consumption and the number of units are the same with the others in the literature. How-

Table 6. Optimal results for case 3

Method	No. of units	q^{cu} (kW)	Annual cost ($)
Linnhoff and Flower [38]	10	1975	43934
Lewin [14]	10	1879	43799
Yerramsetty and Murty [19]	10	1879	43538
Lewin et al. [15]	10	1879	43452
Pariyani et al. [22]	10	1879	43439
Gupta and Ghosh [23]	10	1879	43342
Lin and Miller [20]	10	1879	43329
This work	10	1879	43431

Table 7. Problem data for case 4

Stream	T_{in} (K)	T_{out} (K)	W (kW/K)
H1	358	318	156.3
H2	393	313	50
H3	398	308	23.9
H4	329	319	1250
H5	363	359	1500
H6	498	348	50
C1	313	328	466.7
C2	328	338	600
C3	338	438	180
C4	283	443	81.3
Steam	200	198	-
Water	288	293	-

U=0.025 kW/m^2·K for all matches; Annual cost ($)=60A for all exchangers; Chu=100 $/kW·year; Ccu=15 $/kW·year

Table 8. Optimal results for case 4

Method	No. of units	Total area (m^2)	q^{hu} (MW)	q^{cu} (MW)	Annual cost (M$)
Ahmad [34]	-	-	15.4	9.796	7.074
Ravagnani et al. [16]	13	56600	20.529	14.924	5.672
Yerramsetty and Murty [19]	13	56085	20.745	15.14	5.666
Khorasany and Fesanghary [26]	12	58197	19.605	14	5.662
This work	13	57374	19.991	14.385	5.657

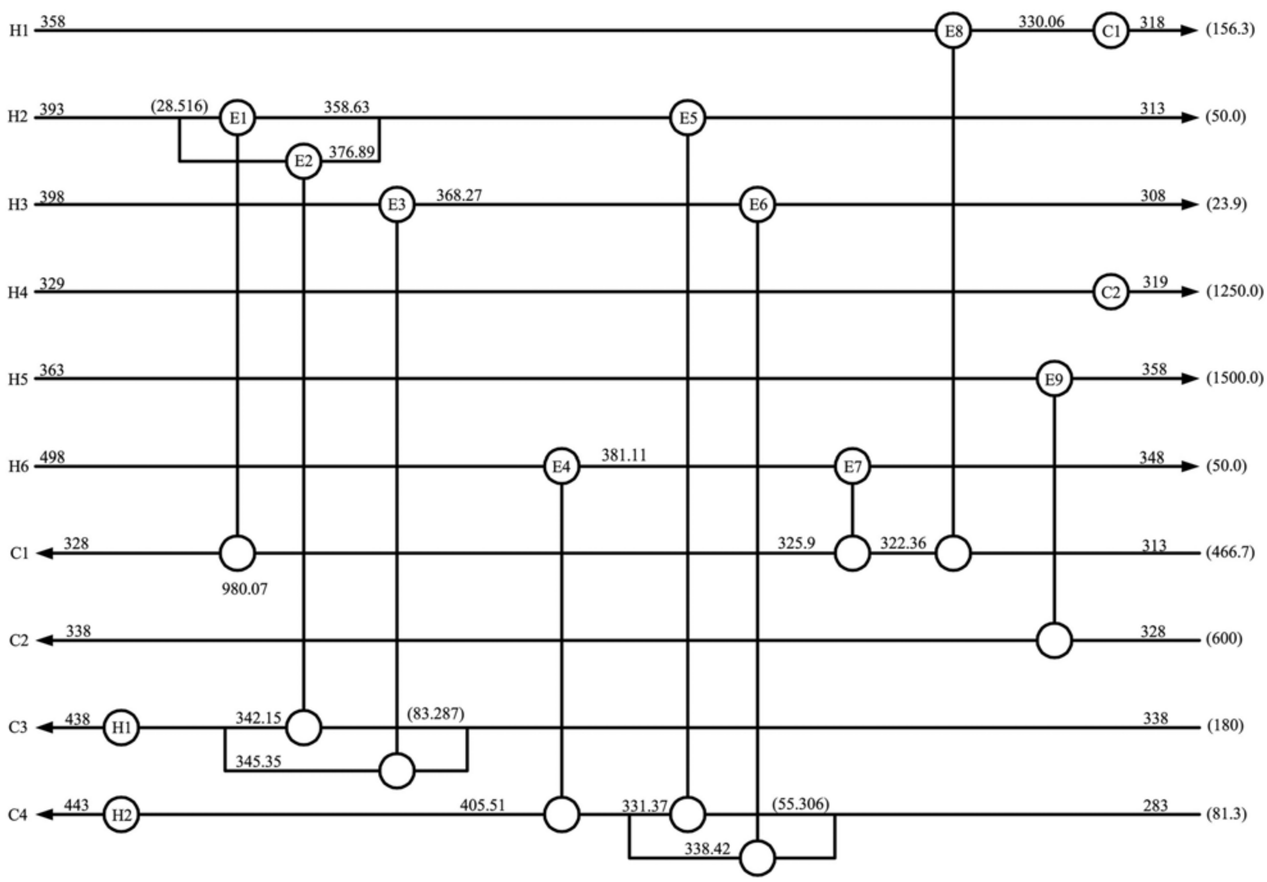

Fig. 8. Optimum HEN structure for case 4.

ever, the network structure obtained in this paper is different from all the reported ones.

4. Case 4

This problem, including six hot streams and four cold streams, has been studied through different methods. Ahmad first obtained an optimal network with annual cost of 7074000 using pinch analysis [34]. Ravagnani et al. found a much better solution of 5672821 using pinch analysis with genetic algorithms [16]. Khorasany and Fesanghary [26] and Yerramsetty and Murty [19] found excellent results during their researches. Table 7 lists the data of this case. The results of this paper as well as those in the literature are summarized in Table 8. The optimal network is shown in Fig. 8. It is obvious that the total annual cost of this configuration is less than that reported in the literature.

ALGORITHM ANALYSIS

The results calculated in section 4 demonstrate the effectiveness of the proposed algorithm in finding better solutions for HENS problems. To guarantee the computational efficiency, there are several parameters of the proposed algorithm that need to be adjusted. In IPSO, the following parameters need to be regulated: initial and final values of the inertial weight, w_{max} and w_{min}; boundary values of acceleration coefficients c_{1s}, c_{1e}, c_{2s} and c_{2e}; maximum iteration number, $iter_{max}$. In GA, the following parameters need to be regulated: crossover rate p_c; mutation rate p_m; maximum iteration number $miter_{max}$. These parameters are derived empirically through numerous experiments, and their values that are finally adopted in the implementation are listed in Table 9.

The population size of the proposed algorithm generally depends on the complexity of the problems. Although a larger population size will significantly increase the computation time, it is preferred for complex cases. To save computation time, the complexity of the problems can be reduced by decreasing the numbers of stages and branches for each stream. Empirically, the number of stages of the optimum structure is seldom larger than the number of hot/cold streams, and the number of branches is usually not larger than 2. Fifty independent runs for four cases studied are executed to test the stability of the proposed algorithm. We can conclude from Table 10 that the best solutions of all of the four cases can be found with a high probability. The convergence behavior of the proposed algorithm for Case 1 and Case 3 is plotted in Fig. 9. It is worthwhile to note that relative amounts of CPU time were taken in the Cases 2-4. Apparently, in the stage-wise simultaneous HENS, the computation time increases with the increasing of the number of process streams, the maximum number of stream splits allowed, and the number of stages appointed. In our work, Cases 2-4 are almost the same in the above respects. However, case 3 requires much smaller time than case 2 and case 4. It is chiefly because the proposed algorithm is not a computationally efficient exact solution algorithm. Moreover, the complexity of HENS problems is not always readily apparent upon inspection. These results should provide motivation for the further research of efficient global solution technique and computational complexity analysis.

Table 9. Parameters of the proposed algorithm

GA	Value	IPSO	Value
p_c	0.6	w_{max}	0.9
p_m	0.005	w_{min}	0.4
$iter_{max}$	100	c_{1s}	2.5
		c_{1e}	0.5
		c_{2s}	0.5
		c_{2e}	2.5
		$miter_{max}$	50

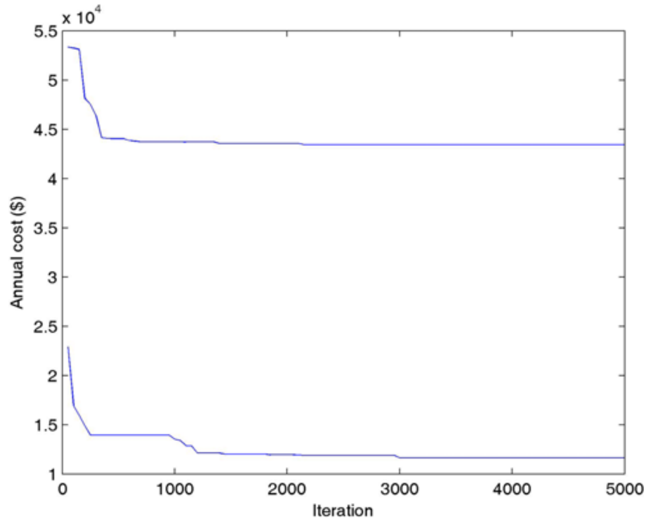

Fig. 9. Convergence behavior of the proposed algorithm.

CONCLUSIONS

An efficient two-level optimization method has been used in the present work for HENS. The HENS problem, considering splitting streams and non-isothermal mixing, is usually formulated as a non-

Table 10. The probability of obtaining the best solution and the computation time

Parameters	Case 1	Case 2	Case 3	Case 4
Population size of GA	50	100	100	100
Population size of IPSO	50	50	50	50
Number of stages	3	3	3	3
Maximum number of branches	1	2	2	2
Average CPU time	80 s	4342 s	325 s	5432 s
Probability of best solution obtained	100%	80%	92%	70%

convex MINLP model. In the proposed algorithm, discrete and continuous variables are updated independently according to a given criterion. GA is employed to implement structural optimization in the upper level, then stream split flows and heat exchanger duties are optimized for the given HEN structure in the lower level by IPSO algorithm. The constraints handling strategy based on salient feature of HENS problem is used to enhance the performance of the IPSO. Four well-known benchmark cases from other literature have been studied, and the results show that the proposed method is effective in finding better solutions than those most reported in the literature.

NOMENCLATURE

Indices

c : cold process stream
h : hot process stream
i : index for hot process stream
in : inlet
j : index for cold process stream
k : index for stage
out : outlet

Parameters

c_1, c_2 : cognitive and social acceleration coefficients
c_{1s}, c_{1e}, c_{2s}, c_{2e} : initial and finial cognitive and social acceleration coefficients with time varying
C_{ef} : fixed charge for heat exchanger unit
C_{ea} : area cost coefficient for heat exchanger
C_{cf} : fixed charge for cooler unit
C_{ca} : area cost coefficient for cooler
C_{hf} : fixed charge for heater unit
C_{ha} : area cost coefficient for heater
C_{cu} : per unit cost for cold utility
C_{hu} : per unit cost for hot utility
EMAT : minimum fixed temperature approach for each match.
He : exponent for area cost of heat exchanger
Hcu : exponent for area cost of cooler
Hhu : exponent for area cost of heater
r_1, r_2 : uniform random numbers between 0 and 1
$iter_{max}$: maximum number of iterations
t : current iteration
$T^h_{i,in}$: initial inlet temperature of hot stream i [K]
$T^h_{i,out}$: final outlet temperature of hot stream i [K]
$T^c_{j,in}$: initial inlet temperature of cold stream j [K]
$T^c_{j,out}$: final outlet temperature of cold stream j [K]
T_heat_in : inlet temperature of hot utility [K]
T_heat_out : outlet temperature of hot utility [K]
T_cool_in : inlet temperature of cold utility [K]
T_cool_out : outlet temperature of cold utility [K]
$U_{i,j}$: overall heat transfer coefficient for match of hot stream i and cold stream j [$kW/m^2\cdot K$]
$U_{hu,j}$: overall heat transfer coefficient for match of cold stream j and hot utility [$kW/m^2\cdot K$]
$U_{cu,i}$: overall heat transfer coefficient for match of hot stream i and cold utility [$kW/m^2\cdot K$]
W^h_i : heat capacity of hot stream [kJ/K]
W^c_j : heat capacity of cold stream [kJ/K]
w_{max}, w_{min} : the maximum and minimum inertia

Variables

LMTD : log mean temperature difference
$A^e_{i,j,k}$: area for match of hot stream i and cold stream j in stage k [m^2]
A^{cu}_i : area for match of hot stream i and cold utility [m^2]
A^{hu}_j : area for match of cold stream j and hot utility [m^2]
$Be_{i,j,k}$: binary variable to denote existence for match of hot stream i and cold stream j in stage k
Bcu_i : binary variable to denote existence for match of hot stream i and cold utility
Bhu_j : binary variable to denote existence for match of cold stream j and hot utility
$q_{i,j,k}$: heat exchanged between hot stream i and cold stream j in stage k [kW]
q^{hu}_j : heat exchanged between cold stream j and hot utility [kW]
q^{cu}_i : heat exchanged between hot stream i and cold utility [kW]
rand : random value between 0 and 1
$t^h_{i,k,in}$: inlet temperature of hot stream i in stage k [K]
$t^h_{i,k,out}$: outlet temperature of hot stream i in stage k [K]
$t^c_{j,k,in}$: inlet temperature of cold stream j in stage k [K]
$t^c_{j,k,out}$: outlet temperature of cold stream j in stage k [K]
$tp^h_{i,j,k,in}$: inlet temperature of branching hot stream for match of hot stream i and cold stream j in stage k [K]
$tp^h_{i,j,k,out}$: outlet temperature of branching hot stream for match of hot stream i and cold stream j in stage k [K]
$tp^c_{i,j,k,in}$: inlet temperature of branching cold stream for match of hot stream i and cold stream j in stage k [K]
$tp^c_{i,j,k,out}$: outlet temperature of branching cold stream for match of hot stream i and cold stream j in stage k [K]
$Wp^h_{i,j,k}$: heat capacity of branching hot stream for match of hot stream i and cold stream j in stage k [K]
$Wp^c_{i,j,k}$: heat capacity of branching cold stream for match of hot stream i and cold stream j in stage k [K]
Pbest : the best position of the each particle
Gbest : the best position of swarm
v_{id} : the velocity vector for the particle i
x_{id} : the position vector for the particle i
Q_{max} : an upper bound for heat exchanger [kW]

REFERENCES

1. K. C. Furman and N. V. Sahinidis, *Ind. Eng. Chem. Res.*, **41**(10), 2335 (2002).
2. K. C. Furman and N. V. Sahinidis, *Comput. Chem. Eng.*, **25**(9/10), 1371 (2001).
3. S. H. Choi and V. Manousiouthakis, *Korean J. Chem. Eng.*, **19**(2), 227 (2002).
4. I. E. Grossmann and R. W. H. Sargent, *Ind. Eng. Chem. Process Des. Dev.*, **18**(2), 343 (1979).
5. J. Viswanathan and I. E. Grossmann, *Comput. Chem. Eng.*, **14**(7), 769 (1990).
6. M. A. Durn and I. E. Grossmann, *Mathematical Programming*, **36**(3), 307 (1986).
7. H. S. Ryoo and B. P. Sahinidis, *Comput. Chem. Eng.*, **19**(5), 551

(1995).
8. T. Westerlund and F. Petersson, *Comput. Chem. Eng.*, **19**(suppl.1), 131 (1995).
9. T. F. Yee and I. E. Grossmann, *Comput. Chem. Eng.*, **14**(10), 1165 (1990).
10. J. M. Zamora and I. E. Grossmann, *Comput. Chem. Eng.*, **22**(3), 367 (1998).
11. K. M. BjÖrk and T. Westerlund, *Comput. Chem. Eng.*, **26**(11), 1581 (2002).
12. M. F. Cardoso, R. L. Salcedo, S. Feyo de Azevedo and D. Barbosa, *Comput. Chem. Eng.*, **21**(12), 1349 (1997).
13. S. H. Choi, J. W. Ko and V. Manousiouthakis, *Comput. Chem. Eng.*, **123**(9), 1351 (1999).
14. D. R. Lewin, H. Wang and O. Shalev, *Comput. Chem. Eng.*, **22**(10), 1503 (1998).
15. D. R. Lewin. *Comput. Chem. Eng*., **22**(10), 1387 (1998).
16. M. A. S. S. Ravagnani, A. P. Silva, P. A. Arroyo and A. A. Constantino, *Appl. Therm. Eng.*, **25**(7), 1003 (2005).
17. T. Laukkanen and C. J. Fogelholm, *Comput. Chem. Eng.*, **35**(11), 2389 (2011).
18. R. Lotif and R. B. Boozarjomehry, *Ind. Eng. Chem. Res.*, **49**(10), 4731 (2010).
19. K. M. Yerramsetty and C. V. S. Murty, *Comput. Chem. Eng*., **32**(8), 1861 (2008).
20. B. Lin and D. C. Miller, *Comput. Chem. Eng.*, **28**(8), 1451 (2004).
21. S. Chakraborty and P. Ghosh, *Chem. Eng. J.*, **72**(3), 209 (1999).
22. A. Pariyani, A. Gupta and P. Ghosh, *Comput. Chem. Eng.*, **30**(6/7), 1046 (2006).
23. A. Gupta and P. Ghosh, *Comput. Chem. Eng.*, **34**(10), 1632 (2009).
24. S. J. Wu and P. T. Chow, *Comput Struct.*, **56**(6), 979 (1995).
25. X. Luo, Q. Y. Wen and G. Fieg, *Comput. Chem. Eng.*, **33**(6), 1169 (2009).
26. R. M. Khorasany and M. Fesanghary, *Comput. Chem. Eng.*, **33**(8), 1363 (2009).
27. H. G. Dong, C. Y. Lin and C. T. Chang, *Chem. Eng. Res. Des.*, **86** (A3), 299 (2008).
28. J. Kennedy and R. Eberhart, *Proceedings of IEEE International Conference on Neural Networks Neural Network*, VI. IEEE, Service Center, Piscataway, NJ, USA (1995).
29. J. Eberhart and Y. Shi, *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea (2001).
30. T. F. Yee, I. E. Grossmann and Z. Kravanja. *Comput. Chem. Eng.*, **14**(10), 1151 (1990).
31. J. J. J. Chen, *Chem. Eng. Sci.*, **42**(10), 2488 (1987).
32. A. Ratnaweera, S. K. Halgamuge and H. C. Watson, *IEEE. T. Evolut. Comput.*, **8**(3), 240 (2004).
33. Y. Shi and R. Eberhart, *Proceedings of IEEE international conference of evolutionary computation*, Piscataway, New York (1998).
34. S. Ahmad. Ph.D. Thesis, UMIST, Manchester, UK (1985).
35. J. S. Nielsen, M. W. Hansen and S. B. Joergensen, *Comput. Chem. Eng.*, **20**(suppl.1), 249 (1996).
36. B. Linnhoff and S. Ahmad, *Comput. Chem. Eng.*, **14**(7), 729 (1990).
37. F. Pettersson, *Comput. Chem. Eng.*, **29**(5), 993 (2005).
38. B. Linnhoff and J. R. Flower, *AIChE J.*, **24**(4), 642 (1978).

COMMUNICATIONS

Our work aims to present an efficient and stable algorithm for heat exchanger network synthesis. An improved particle swarm optimization algorithm and a genetic algorithm are employed to perform the structural and parametric optimization separately. Handling strategy of constraints and appropriate simplification of structures are also discussed for improving the probability of obtaining desired optimum solution and saving computation time.