Chemical Engineering Journal 72 (1999) 209–216

Heat exchanger network synthesis: the possibility of randomization

Samarjit Chakraborty[1,2,a], Pallab Ghosh[b,*]

[a]*Institut für Technische Informatik und Kommunikationsnetze, Eidgenössische Technische Hochschule, Zürich CH-8092, Switzerland*
[b]*Department of Chemical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400 076, India*

Received 10 February 1998; received in revised form 6 November 1998; accepted 14 November 1998

Abstract

This paper presents a new approach to heat exchanger network (HEN) design making extensive use of randomization techniques. It is exceedingly simple to implement and gives new insight into the hardness and the cost landscape underlying a given problem. At the same time, the results from our algorithm may be used as good initial solutions required by most non-linear optimization problem formulations of HEN design. Practical networks involve trade-off between a number of factors, all of which are difficult to incorporate in a single design methodology. Our approach is blind to any design heuristic and generates a sufficiently large number of networks that can be further evaluated to pick up the most suitable network depending on specific design requirements. However, the current version of the algorithm is limited to HEN synthesis problems that can be solved without stream splitting. We have experimented with the three standard literature problems and obtained results that compare well with the previously published results, which justify further research in this direction.

Keywords: Heat exchanger network design; Randomized algorithms; Randomized search

1. Introduction

Synthesis of an optimal heat exchanger network (HEN) with any one of the targets like minimization of utility, the number of exchangers, or the annual cost, to bring each process stream from its inlet to target temperature, is a combinatorial optimization problem. Although one of the most studied problems of process synthesis over the last three decades, it is still open to further research. The problem was first proposed by Masso and Rudd [1] in 1969 and since then many design algorithms have been proposed, an extensive review of which can be found in [2].

In the last decade, computer science has witnessed a tremendous growth in the area of randomized algorithms [3]. A randomized algorithm is one that receives, in addition to its input data, a stream of random bits that it can use for the purpose of making random choices during its course of execution. Thus, different runs of the algorithm may give different results and even for a fixed input the execution time might be a random variable. It is now recognized that in a wide range of applications, randomization is an important tool for the construction of algorithms. It went from being a tool in computational number theory to finding wide spread application in areas starting from pattern matching, sorting and searching, computational geometry, graph theory and data structure maintenance, to combinatorial enumeration and distributed computing.

There are two main advantages that randomization often leads to. Firstly, often the execution time or space requirement becomes smaller than that of the best known deterministic algorithm for the problem. Secondly, the several randomized algorithms invented so far are invariably extremely simple to understand and implement. Often, the introduction of randomization suffices to convert a simple deterministic algorithm with worst case behavior into a randomized algorithm that performs well with high probability on every possible input.

The aim of this paper is to explore the possibility of using randomization in HEN design. Although randomization has been used previously in this field, in the form of genetic algorithms [4] and simulated annealing [5], this paper, to the best of our knowledge, represents the first work where it has been used in such a stark way.

The randomized approach that we present, derives its motivation from a technique used by Finn et al. [6] to solve a

*Corresponding author. E-mail: pallab@che.iitb.ernet.in
[1]This work was carried out when the author was at the Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur.
[2]samarjit@tik.ee.ethz.ch

1385-8947/99/$ – see front matter
PII: S1385-8947(99)00007-8

conformational search problem to produce a number of distinct low energy conformations of a given drug molecule. They used a randomized approach where each degree of freedom of the molecule is randomly chosen from a given space, followed by a minimization and clustering. The main idea being that, a systematic procedure has a higher chance of missing the irregularly shaped basins of attraction of the energy landscape of the molecule.

We have presented three standard literature problems involving 6, 9 and 10 streams. In all the cases we have obtained results that are close to the published results and in some cases even better than known results. Section 2 briefly describes previous work on HEN design, followed by our algorithm in Section 3. We report in Section 4, results related to three standard problems. Section 5 discusses the directions for a formal analysis of our approach, some open issues that merit consideration in future work and possible applications of this work, and finally, Section 6 contains the conclusion and implications of this work.

2. Previous work

The supply and removal of heat in a modern chemical process plant represents an important problem in the process design of the plant. The cost of facilities to accomplish the desired heat exchange between the hot and cold media may amount to one third of the total cost of the plant. Thus, a lot of research work has been done to find the optimum configuration of a HEN both in terms of total cost and operability.

One of the most important insights that has been developed to overcome the combinatorial nature of this problem is the prediction of the minimum utility target [7,8], which can be performed prior to developing the network structure. The number of network configurations satisfying the minimum utility target is often much smaller than the total number of possible configurations and furthermore this target ensures that the lowest utility cost will be obtained for a given minimum temperature approach. A recent approach [9] for multiple utility targeting is based upon a cheapest utility principle (CUP) which simply states that it is optimal to increase the load of the cheapest utility and maintain the loads of the relatively expensive utilities constant while increasing the total utility consumption. Therefore, a major problem that remains in the synthesis of HENs is how to generate the network structures that not only satisfy the minimum overall utility cost target but that also feature minimum investment cost as well as fewest numbers of units [10].

Several approaches have been used to solve this problem. The technique based upon thermodynamically established principles, the ''pinch'' method of Linnhoff and Hindmarsh [11] has proved to be exceptionally suitable in many situations and consequently has found wide use. Current versions of the pinch design using ''driving force plot'' and ''remaining problem analysis'' yield improved solutions. Mathematical programming techniques in which the various subtasks of the synthesis problem are automated have also been used quite frequently [12,13]. The mathematical programming framework formulates the HENs as mixed integer non-linear programming (MINLP) models that have continuous variables and integer decisions. Non-linear optimization problems usually require a good initial solution and often convergence to global minimum is not ensured. The good starting points required in these models are seldom available. If a poor guess is provided, the problem may converge to an inferior solution or even may fail to converge. Another drawback is that, the mathematical formulations in these algorithms are often based upon a large number of small intervals (either temperature intervals or enthalpy intervals or both). Thus, even for moderately sized problems the number of intervals may be quite large making the corresponding optimization problem to grow to an unmanageable size. The ''spaghetti'' design concepts of Townsend and Linnhoff [14], the ''stage'' concept of Yee and Grossmann [15] or the ''block'' concepts of Zhu et al. [16] have been used in recent years to reduce the size of the problem. However, the ''spaghetti'' design usually requires a large number of exchangers as a result of matching hot and cold streams in each section of the composite curves to obtain vertical heat transfer. The design may be simplified by setting the number of stages to a fixed value (usually not greater than the maximum number of hot and cold streams) [15]. But even after applying these techniques, large industrial systems may still generate optimization problems of huge size and complexity.

Another problem arises out of the uncertainties that arise out of the non-convexities in the network optimization problem. A global optimum search has been proposed by Floudas and Ciric [17] to decompose the non-convex network optimization problem into a set of convex sub-problems that represent upper and lower bounds and whose solution can lead to the network configuration with globally minimum investment cost. Recently, Zhu et al. [18] have coupled their ''block'' method to synthesize HENs featuring streams with unequal film heat transfer coefficients. It also utilizes the concept of individual stream ΔT contributions introduced by Rev and Fonyo [19]. Potentially good matches are readily identified using these techniques and topology traps are avoided. As a consequence, good initial designs are generated and available for subsequent cost optimization using non-linear programming. The chances of finding a globally optimum network are thus increased. These heuristic methods do not rely upon pinch decomposition.

Recently, non-traditional optimization techniques like simulated annealing [5] and genetic algorithms [4] are being used to obtain globally optimum HENs. The simulated annealing technique is based upon the mathematical theory of Markov chains. Two aspects of the algorithm strongly affect the computation time: the evaluation of the change in cost between different randomly generated states and the annealing temperature schedule. The work of Dolon et al. [5] presents a new implementation of the simulated annealing algorithm treating the first aspect efficiently through the use of a linked tree data structure to calculate changes in cost directly resulting in a speed increase of two orders of magnitude over the earlier implementations of the same algorithm.

On the other hand, searches based upon the genetic algorithms (GAs) use the principles of natural selection and natural genetic systems. GAs are empirically and to some extent theoretically proven to provide robust search in complex spaces even if the objective function is not continuous or smooth. They may be suitable candidates to solve large combinatorial optimization problems like the HEN synthesis.

3. Randomized approach

The main idea behind our approach is to randomly sample points from the space consisting of all possible networks that satisfy the heat load constraints dictated by the problem. This reduces to constructing random networks for the given problem. The only assumption that we make for constructing these networks is, for each stream, the number of exchangers through which it passes is bound by a constant. This constant should usually be different for different streams depending on the total amount of heat it should exchange, but has to be predefined by the designer. We feel that this is not a serious constraint from a practical point of view since for all practical networks, this is a very low value and an upper bound can be very easily guessed for any problem.

The present version of our algorithm does not support stream splitting. Further investigation in this direction is required to address this issue. Considering the present state of art in heat exchanger design this appears to be a serious drawback; we discuss the consequences of this in Section 5.

First we give an overview of our algorithm and then formally state it. Conceptually it consists of two stages – determining the network topology and heat load distribution. In both the stages we completely rely on randomized techniques. In the formal statement of the algorithm the two stages are overlapped. For each stream we assign a number of exchangers equal to the constant mentioned in the first paragraph. These exchangers are ordered according to the sequence in which the stream enters them. Now from this set, one exchanger is randomly chosen for each stream and kept aside, the reason for which will become apparent as we progress. From the remaining, each exchanger belonging to a hot stream is randomly mapped to an exchanger belonging to a cold stream and with a small probability is mapped to a cold utility. The unmapped exchangers belonging to cold streams are now mapped to hot utility. The procedure could as well have been reversed by starting with exchangers belonging to cold streams. This completes the structure determination part. Now in the generated network, for each exchanger, we assign a heat load based on the outcome of a random choice over three different possibilities, each with a different probability. No heat load is assigned at all, in which case the exchanger becomes vacuous; a randomly chosen value from the allowed range is assigned; the third possibility being that, the maximum value from the allowed range is assigned. Lastly, if still some amount of heat remains to be exchanged for any stream, it is done by assigning the exchanger that was kept aside, to a utility. Now we formally state the algorithm.

For each stream i, let the maximum number of exchangers through which this stream can pass be bounded by N_i. Let the exchangers through which stream i passes be $E_i = \{e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}\}$.

Let H denote the set of exchangers through which a hot stream passes and C denote the set of exchangers through which a cold stream passes.

$$H = \bigcup_{i \text{ is a hot stream}} E_i$$

$$C = \bigcup_{i \text{ is a cold stream}} E_i$$

Clearly, H and C are not disjoint unless all the hot streams are matched with cold utilities and all cold streams with hot utilities. Further, $q(e_{ij})$ denotes the amount of heat exchanged in the exchanger e_{ij}.

3.1. Algorithm

//Keep aside a randomly chosen exchanger for each stream i, which will be assigned to a utility in case some amount of heat still remains to be transferred to or from i, after the heat load distribution for it is over//

for each stream i
{
Let r_i be a randomly sampled element of E_i, without replacement;
$H \leftarrow H \backslash \{r_i\}$;
$C \leftarrow C \backslash \{r_i\}$;
$Q_i \leftarrow$ total amount of heat that has to be transferred to (or from) i;
}
for each $e \in \cup_{i=1}^{N} E_i$, where $N =$ total number of streams
$q(e) \leftarrow 0$;

//Each exchanger through which a hot stream passes is either mapped to a randomly chosen exchanger through which a cold stream passes, or is mapped to a cold utility, or is left vacuous. Each of this is done with a specified probability.//

for each $e_{ij} \in H$
{
Define a probability distribution over the steps 1,2,3,4;

Execute any one of the steps in accordance with the probability distribution;
Step 1: $\{q(e_{ij}) \leftarrow 0\}$;
Step 2: {Let e_{kl} be a randomly sampled element of C, without replacement;

Map e_{ij} to e_{kl} and vice-versa;
$q(e_{ij}) \leftarrow q(e_{kl}) \leftarrow$ a randomly chosen value in the interval [0, min(Q_i, Q_k)];
$Q_i \leftarrow Q_i - q(e_{ij})$;
$Q_k \leftarrow Q_k - q(e_{kl})$;}

Step 3: {Let e_{kl} be a randomly sampled element of C, without replacement;

Map e_{ij} to e_{kl} and vice-versa;
$q(e_{ij}) \leftarrow q(e_{kl}) \leftarrow \min(Q_i, Q_k)$;
$Q_i \leftarrow Q_i - q(e_{ij})$;
$Q_k \leftarrow Q_k - q(e_{kl})$;}

Step 4: {Map e_{ij} to cold utility;

$q(e_{ij})$ = a uniformly chosen value in the interval $[0, Q_i]$;
$Q_i \leftarrow Q_i - q(e_{ij})$;}

}

//For each hot stream i in case some amount of heat still remains to be transferred from i, adjust it with a cold stream or map it to a cold utility//

for each hot steam i
if $Q_i > 0$
{
for each $e_{ij} \in H$
{// For each exchanger through which stream i passes//
if e_{ij} is mapped to a cold utility or $q(e_{ij}) = 0$
{
$q(e_{ij}) \leftarrow q(e_{ij}) + Q_i$;
$Q_i \leftarrow 0$;
map e_{ij} to cold utility;
}
else
{
if e_{ij} is mapped to e_{kl}
{
//Increase the heat exchange between the streams i and k, if possible//
$q \leftarrow \min(Q_i, Q_k)$;
$q(e_{ij}) \leftarrow q(e_{ij}) + q$;
$q(e_{kl}) \leftarrow q(e_{kl}) + q$;
$Q_i \leftarrow Q_i - q$;
$Q_k \leftarrow Q_k - q$;
}
}
}
//If some amount of heat still remains to be exchanged from stream i, adjust it by mapping the exchanger r_i to cold utility//
if $Q_i > 0$
{
map r_i to cold utility;
$q(r_i) \leftarrow Q_i$;
$Q_i \leftarrow 0$;>
}
}

//For each cold stream k, if some amount of heat still remains to be transferred to k, map it to a hot utility//
for each cold stream k
if $Q_k > 0$
{
for each $e_{kl} \in C$// For each exchanger through which stream k passes//
if $q(e_{kl}) = 0$
{
$q(e_{kl}) \leftarrow Q_k$;
$Q_k \leftarrow 0$;
map e_{kl} to hot utility;
}
if $Q_k > 0$
{
map r_k to hot utility;
$q(r_k) \leftarrow Q_k$;
$Q_k \leftarrow 0$;
}
}

4. Results

We have experimented with three problems involving 6 (6SP) [20], 9 (9SP) [21] and 10 (10SP) [20] streams. Details of the problems can be found in the Appendices A, B and C. In all the cases 50 000 random networks were generated, and the upper bound on the number of exchangers assigned to each stream was set to 4.

The probabilities with which steps 1, 2, 3 and 4 were executed, were 0.2, 0.3, 0.4 and 0.1, respectively. Tables 1–3 present the results obtained for the three problems for 10 different sequences of random numbers. Table 4 shows the statistics of CPU time required for generating 50 000 networks, obtained over 10 different runs on a moderately loaded DEC Alpha 2000 4/233 (Dual Processor system) server running Digital Unix with 128 MB of RAM.

Figs. 1–3 show some of the results reported in Tables 1–3.

5. Discussion

In the HEN design problem, since it is difficult to systematically explore the underlying search space without the help of some heuristic, a randomized approach is a natural alternative. Still it is difficult to justify the use of our algorithm as a stand alone technique for this problem, mostly because all it does is simply a random search.

Table 1
Results obtained for 6SP (C_{opt}=$35 010 [20])

Run	Minimum cost network obtained ($)	Number of networks with cost $\leq(1+\varepsilon)\ C_{opt}$	
		ε=0.1	ε=0.2
1	35407.94	80	83
2	35407.93	81	86
3	35407.94	73	75
4	35415.04	95	101
5	35407.94	87	93
6	**35016.06**	76	80
7	35407.94	79	81
8	35407.93	84	90
9	35427.25	83	87
10	35407.94	88	89

Table 2
Results obtained for 9SP (C_{opt}=$\$4.23\times10^6$ [21])

Run	Minimum cost network obtained ($)	Number of networks with cost $\leq(1+\varepsilon)\ C_{opt}$	
		ε=0.1	ε=0.2
1	3038454.25	4	10
2	**2970280.03**	1	5
3	3092870.50	1	7
4	3052415.25	1	6
5	3073271.00	2	4
6	3073493.75	1	5
7	3092870.50	2	6
8	3091955.25	2	5
9	3002199.75	1	6
10	3073213.75	2	5

Table 3
Results obtained for 10SP (C_{opt}=$43 984 [20])

Run	Minimum cost network obtained ($)	Number of Networks With Cost $\leq(1+\varepsilon)\ C_{opt}$	
		ε=0.1	ε=0.2
1	45030.33	1	1
2	44768.79	1	1
3	44768.79	1	1
4	45167.51	1	1
5	**44213.88**	1	1
6	44668.77	1	1
7	44768.79	1	1
8	44752.72	1	1
9	45030.33	1	1
10	45167.51	1	1

Table 4
CPU time required for generating 50 000 random networks

Problem	Minimum (s)	Maximum (s)	Average (s)
6SP	5.13	5.28	5.23
9SP	6.28	6.36	6.34
10SP	18.22	18.38	18.34

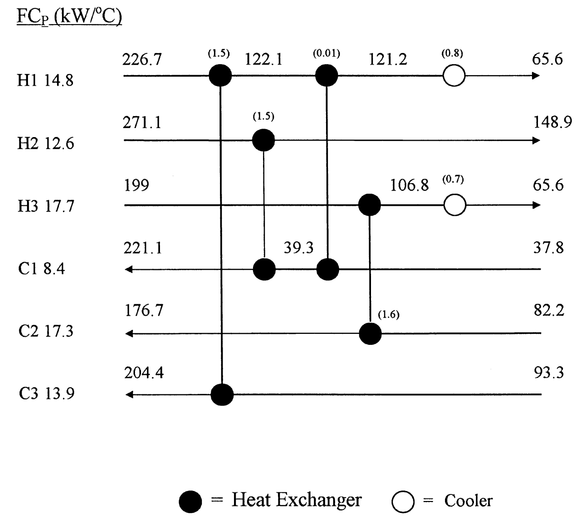

Fig. 1. Optimum network obtained for 6SP. This network is same as that reported in [20]. Cost=35016.06 $/year. The heat capacity flow rates (FC_P, kW/°C) of the streams are shown in the figure. The heat loads (MW) for each exchanger are shown in the parentheses. The stream temperatures shown are in °C.

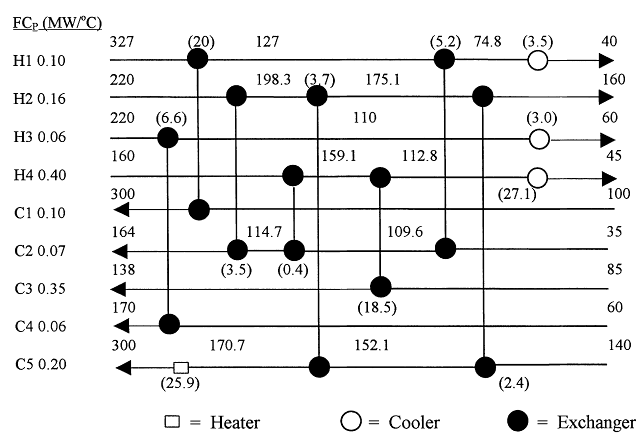

Fig. 2. Optimum network obtained for 9SP. Cost=2970280.03 $/year. The heat capacity flow rates (FC_P, MW/°C) of the streams are also shown in the figure. The heat loads (MW) for each exchanger are shown in the parentheses. The stream temperatures shown are in °C.

However, we feel that there are situations where it can be directly used in the form presented here, in conjunction with other methods. Most non-linear optimization problem formulations of the HEN problem require a good initial solution, which is seldom available. The results generated by our algorithm can be of direct use here. Again, the non-traditional optimization techniques like genetic algorithms and simulated annealing also require starting solutions and thus can be interfaced with our formulation very effectively. Moreover, practical networks involve trade-off between number of exchangers, area, energy and ease of operability, all of which is difficult to incorporate in a single design methodology. Our approach is blind to all of these. Thus, a number of networks generated by this algorithm can be further evaluated and the most suitable chosen out of them by manual inspection.

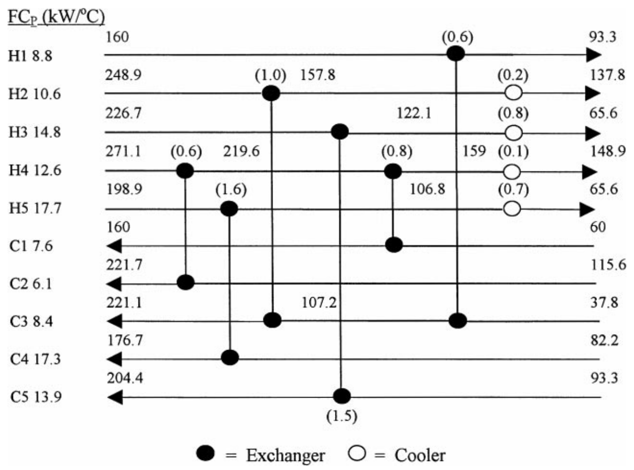

Fig. 3. Optimum network obtained for 10SP. Cost=44213.88 \$/year. The heat capacity flow rates (FC_P, kW/°C) of the streams are also shown in the figure. The heat loads (MW) for each exchanger are shown in the parentheses. The stream temperatures shown are in °C.

The results presented in Section 6 show a wide variation in the different runs for some problems. Thus from a completely different perspective, this technique can serve as a tool to study the cost landscape underlying any heat exchanger problem, which might give new insight in formulating other design methodologies. Considering the nature of the problem, it is difficult to formally analyze the behavior of the algorithm. Clearly, for a random search to succeed, there must be an abundance of good solutions. A possible direction might be as follows: given an optimal network, if we can show that a slight perturbation results in a network whose cost also changes by a small amount, it would imply that there exists a lot of networks having near optimal costs and hence justify a random search.

For the 9SP problem, it has been reported in literature [21] that the higher temperature pinch occurs at $\Delta T_{\min}$= 10°C and the lower temperature pinch occurs at $\Delta T_{\min}$= 26°C. The pinch swap takes place when $\Delta T_{\min}$=19°C. Thus, a ''topology trap'' is said to exist since a given set of streams around the pinch can be classified as belonging to the same ''topology region'' and networks from one topology region may not be evolved into another. As a result of this, it is not possible to evolve from the higher cost configurations said to be existing with $\Delta T_{\min}$=10°C to much lower cost networks reported to be existing at $\Delta T_{\min}$=26°C. However, in our procedure, several lower cost networks are seen to exist at $\Delta T_{\min}$=10°C without taking into account the restrictions imposed by the pinch design [21].

To use this method independently, it might be useful to use concepts similar to those presented in [6] for molecular conformational search. After each network is randomly generated, an efficient minimizer [22,23] can be used to transform it to a local minimum nearest to it. This way, by generating networks uniformly over the underlying space, the chances of finding the optimal network can be expected to increase dramatically.

As mentioned in Section 3, our algorithm in the present form does not support stream splitting. However, the results obtained with the three sample problems compare well with the results obtained by other methods which support stream splitting. Our emphasis in this work has been to design an algorithm as simple as possible. To this effect, we have also not used any checking for the minimum approach temperature violation. However, the results presented in the previous section lists only those networks which satisfy this constraint. There might have been other lower cost networks which violate the minimum allowable approach temperature, but we have not considered them. It would be interesting to study how the algorithm behaves after stream splitting is incorporated. The source code implementing our algorithm is available with the first author and may be used to carry out further research in this direction.

6. Conclusion

This paper represents an attempt to attract the chemical engineering community to consider randomization as a possible tool for heat exchanger network design. The approach presented here is a randomized search of the space consisting of all possible networks. The results obtained with the three sample problems presented here strengthen the now widely believed fact that for searching high-dimensional spaces, randomized exploration is superior to systematic exploration when the shape of the underlying space is irregular [24]. It would be interesting to formally analyze the scope of this approach and to explore problems which might be pathological to this type of random search. Moreover, it would be instructive to examine ways to incorporate stream splitting into this formulation and also use more problem specific knowledge rather than a simple blind search.

7. Nomenclature

a, b	cost parameters
C	set of exchangers through which a cold stream passes
C_i	cold streams (Figs. 1–3)
C_{opt}	optimum cost reported in literature, \$/year
e_i	exchanger belonging to set E_i
E_i	set of exchangers through which stream ''i'' passes
H	set of exchangers through which a hot stream passes
H_i	hot streams (Figs. 1–3)
N	total number of streams
N_i	maximum number of exchangers through which stream ''i'' can pass
$q(e_i)$	amount of heat exchanged in exchanger e_i, watt
Q_i	total amount of heat that has to be transferred to/from stream ''i'', watt
r_i	randomly sampled element of E_i
$\Delta T_{\min}$	minimum approach temperature, °C

Greek symbols

ε cost fraction

Appendix A

Problem 6SP [20]

Stream	Heat capacity flow rate (W/°C)	Temperature (°C)	
		Inlet	Outlet
1	8441.0	37.8	221.1
2	17283.0	82.2	176.7
3	13901.3	93.3	204.4
4	14772.0	226.7	65.6
5	12556.0	271.1	148.9
6	17726.08	199.0	65.6

Steam: pressure: 3.103×10^6 N/m^2; latent heat: 1.785×10^3 kJ/kg; temperature: 235.6°C.

Cooling water: temperature: 37.8°C; heat capacity: 4.187 kJ/kg °C; maximum outlet temperature: 82.2°C.

Minimum allowable approach temperature: heat exchanger: 11.1°C; steam heater: 13.9°C; water cooler: 11.1°C.

Overall heat transfer coefficient: heat exchanger: 851.75 W/m^2 °C; steam heater: 1135.66 W/m^2 °C; water cooler: 851.75 W/m^2 °C; equipment downtime: 380 h/year

Network cost parameters (area in m^2): a=1456.3; b=0.6; exchanger capital cost ($): $a\times(\text{area})^b$; annual return rate=0.1; cooling water cost=1.102×10^{-4} $/kg; steam cost=$2.205\times10^{-3}$ $/kg.

Appendix B

Problem 9SP [21]

Stream	Heat capacity flow rate (MW°C)	Heat transfer coefficient (MW/m^2 °C) $\times10^3$	Temperature (°C)	
			Inlet	Outlet
1	0.10	0.50	327	40
2	0.16	0.40	220	160
3	0.06	0.14	220	60
4	0.40	0.30	160	45
5	0.10	0.35	100	300
6	0.07	0.70	35	164
7	0.35	0.50	85	138
8	0.06	0.14	60	170
9	0.20	0.60	140	300

Hot utility (*hot oil*): supply temperature: 330°C; target temperature: 250°C; heat transfer coefficient: 0.5×10^{-3} MW/m^2 °C.

Cold utility (*cooling water*): supply temperature: 15°C; target temperature: 30°C; heat transfer Coefficient: 0.5×10^{-3} MW/m^2 °C.

Cost data: exchanger capital cost ($): 10 000+350×area (m^2); plant lifetime: 5 years; rate of interest: 0%; annual cost unit duty of hot utility: 60 000 ($MW/year). Annual cost of unit duty of cold utility: 6000 ($MW/year)

In this problem we have taken the minimum allowable approach temperature to be 10°C. With minimum approach temperature 26°C, cost reported in [21]=$\$2.93\times10^6$.

Appendix C

Problem 10SP [20]

Stream	Heat capacity flow rate (W/°C)	Temperature (°C)	
		Inlet	Outlet
1	7623.3	60.0	160.0
2	6082.8	115.6	221.7
3	8441.0	37.8	221.1
4	17283.0	82.2	176.7
5	13901.3	93.3	204.4
6	8794.5	160.0	93.3
7	10551.2	248.9	137.8
8	14771.7	226.7	65.6
9	12556.6	271.1	148.9
10	17726.1	198.9	65.6

Steam: pressure: 3.103×10^6 N/m^2; latent heat: 1.785×10^3 kJ/kg; temperature: 235.6°C.

Cooling water: temperature: 37.8°C; heat capacity: 4.187 kJ/kg °C; maximum outlet temperature: 82.2°C.

Minimum allowable approach temperature: heat exchanger: 11.1°C; steam heater: 13.9°C; water cooler: 11.1°C.

Overall heat transfer coefficient: 851.75 W/m^2 °C; steam heater: 1135.66 W/m^2 °C; water cooler: 851.75 W/m^2 °C; equipment downtime: 260 h/year.

Network cost parameters (area in m^2): a=1456.3; b=0.6; exchanger capital cost ($): $a\ (\text{area})^b$; annual return rate=0.1; cooling water cost=1.102×10^{-4} $/kg; steam cost-=$2.205\times10^{-3}$ $/kg.

References

[1] A.H. Masso, D.F. Rudd, AIChE J. 15 (1969) 10.
[2] T. Gundersen, L. Naess, Comput. Chem. Eng. 12 (1988) 503.
[3] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, New York, 1995.
[4] S. Chakraborty, A. Paul, T.K. Dutta, Comput. Chem. Eng. (1996), submitted.
[5] W.B. Dolan, P.T. Cummings, M.D. Le Van, Comput. Chem. Eng. 14 (1990) 1039.

[6] P.W. Finn, L.E. Kavraki, J.C. Latombe, R. Motwani, C. Shelton,S. Venkatasubramanian, A. Yao, RAPID: Randomized Pharmacophor-eIdentification for Drug Design, Proceedings of the 13th Annual ACM Symposium on Computational Geometry, 1997, pp. 324–333.
[7] E.C. Hohman, Ph.D. Thesis, University of Southern Carolina, 1971.
[8] B. Linnhoff, J.R. Flower, AIChE J. 24 (1978) 633.
[9] U.V. Shenoy, A. Sinha, S. Bandyopadhyay, Chem. Eng. Res. Des. (A) 76 (1998) 259.
[10] C.A. Floudas, A.R. Ciric, I.E. Grossmann, AIChE J. 32 (1986) 276.
[11] B. Linnhoff, E. Hindmarsh, Chem. Eng. Sci. 38 (1983) 745.
[12] S.A. Papoulias, I.E. Grossmann, Comput. Chem. Eng. 7 (1983) 707.
[13] A.K. Saboo, M. Morari, R.D. Colberg, Comput. Chem. Eng. 10 (1986) 577.
[14] D.W. Townsend, B. Linnhoff, IChemE Research Meeting, Bath, UK, 1984.
[15] T.F. Yee, I.E. Grossmann, Comput. Chem. Eng. 14 (1990) 1151.
[16] X.X. Zhu, B.K. O'Neill, J.R. Roach, R.M. Wood, Chem. Eng. Res. Des. (A) 73 (1995) 919.
[17] C.A. Floudas, A.R. Ciric, Comput. Chem. Eng. 13 (1989) 1133.
[18] X.X. Zhu, B.K. O'Neill, J.R. Roach, R.M. Wood, Comput. Chem. Eng. 19 (1995) 223.
[19] E. Rev, Z. Fonyo, Chem. Eng. Sci. 46 (1991) 1623.
[20] N. Nishida, Y.A. Liu, L. Lapidus, AIChE J. 23 (1977) 77.
[21] B. Linnhoff, S. Ahmad, Comput. Chem. Eng. 14 (1990) 729.
[22] R. Brent, Ph.D. Thesis, Stanford University, 1971.
[23] D. Pierre, Optimization Theory with Applications, Dover, New York, 1986.
[24] L. Kavraki, Ph.D. Thesis, Stanford University, 1995.