Computers and Chemical Engineering 25 (2001) 1371–1390

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Computational complexity of heat exchanger network synthesis

Kevin C. Furman, Nikolaos V. Sahinidis *

Department of Chemical Engineering, University of Illinois at Urbana-Champaign, 600 South Mathews Avenue, Urbana, IL 61801, USA

Received 6 March 2000; received in revised form 5 February 2001; accepted 5 February 2001

Abstract

Heat exchanger network synthesis (HENS) has been the subject of a significant amount of research over the last 40 years. While significant progress has been made towards solving the problem, its computational complexity is not known, i.e., it is not known whether a polynomial algorithm might exist for the problem or not. This issue is addressed in this paper through a computational complexity analysis.

We prove that HENS is $\mathcal{NP}$-hard, thus refuting the possibility for the existence of a computationally efficient (polynomial) exact solution algorithm for this problem. While this complexity characterization may not be surprising, our analysis shows that HENS is $\mathcal{NP}$-hard *in the strong sense*. Therefore, HENS belongs to a particularly difficult class of hard optimization problems. Further, via restriction to the 3-partition problem, our complexity proofs reveal that even the following simple HENS subproblems are $\mathcal{NP}$-hard in the strong sense: (a) the minimum number of matches target problem, (b) the matches problem with only one temperature interval, uniform cost coefficients, and uniform heat requirements of all cold streams.

These results facilitate the computational complexity analysis of more complex HENS problems and provide new insights to structural properties of the problem. They also provide motivation for the development of specialized optimization algorithms and approximation schemes.

Keywords: Computational complexity; Heat exchanger network synthesis; Optimization algorithms

Nomenclature

Acronyms and abbreviations

EMAT	exchanger minimum approach temperature
HEN	heat exchanger network
HENS	heat exchanger network synthesis
HRAT	heat recovery approach temperature
LMTD	log mean temperature difference

Problems

3PART	3-partition
EVM	extended vertical minimum number of matches
KP	minimum binary Knapsack
M	minimum number of matches-transshipment model
M′	minimum number of matches-transportation model
M1	*M* restricted with $K=1$
M1-1	special case of *M*1

* Corresponding author. Tel.: +1-217-2441304; fax: +1-217-3335052.
E-mail address: nikos@uiuc.edu (N.V. Sahinidis).

PII: S0098-1354(01)00681-0

M1-1D	corresponding decision problem to $M1$-1
M2	special case of $M1$
MN	simultaneous match-network
N	minimum area cost network topology
S	simultaneous HENS
U	minimum utilities cost – transshipment model
U′	minimum utilities cost – transportation model
UMN	nondecomposition-based HENS
UU	unconstrained minimum utilities cost
VM	vertical minimum number of matches

Variables and parameters

b	Knapsack lower bound
B	a number in the 3-partition problem
c_m	utility cost coefficient for heating utility m
c_n	utility cost coefficient for cooling utility n
c_{ij}	cost for heat match between hot stream i and cold stream j
C_i	Knapsack cost coefficient for item i
C_{ij}^{A}	area cost for heat exchanger between hot stream i and cold stream j
$C_i^{\mathrm{A,CU}}$	area cost for heat exchanger between hot stream i and cooling utility
$C_j^{\mathrm{A,HU}}$	area cost for heat exchanger between heating utility and cold stream j
C^{CU}	general utility cost coefficient
C^{HU}	general utility cost coefficient
C_{ij}^{F}	fixed cost for heat exchanger between hot stream i and cold stream j
$C_i^{\mathrm{F,CU}}$	fixed cost for heat exchanger between hot stream i and cooling utility
$C_j^{\mathrm{F,HU}}$	fixed cost for heat exchanger between heating utility and cold stream j
d_j	sum of heat demands for heat sink j
dt_i^{CU}	temperature approach for a match of hot stream i and cooling utility
dt_j^{HU}	temperature approach for a match of heating utility and cold stream j
dt_{ijk}	temperature approach for a match ij prior to stage k
dt_{ij}^1	temperature approach for the inlet for i and outlet for j in the exchanger for i and j
dt_{ij}^2	temperature approach for the inlet for j and outlet for i in the exchanger for I and j
D_{jk}	heat demand for cold process stream j in interval k
$f_{k',k''}^{\mathrm{B},k}$	stream in the superstructure for process stream k flowing from the splitter after the exchanger for k and k' to the mixer preceding the exchanger for k and k''
$f_{k'}^{\mathrm{E},k}$	stream in the superstructure for process stream k flowing through the exchanger in the match between k and k'
$f_{k'}^{\mathrm{I},k}$	stream in the superstructure for process stream k flowing form the initial splitter to the mixer preceding the match between k and k'
$f_{k'}^{\mathrm{O},k}$	stream in the superstructure for process stream k flowing from the splitter after the exchanger for the match between k and k' to the final mixer
F_i^{H}	heat capacity flow rate for hot stream i
F_j^{C}	heat capacity flow rate for cold stream j
h_i	film heat transfer coefficient for source i
h_j	film heat transfer coefficient for sink j
K	number of temperature intervals
L_{ij}	lower bound on heat exchange between source i and sink j
$LMTD_{ij}$	log mean temperature difference between source i and sink j
m	number of heat sources
n	number of heat sinks
N	total number of streams $N=m+n$
$N_{\max}$	maximum number of heat matches allowed
P_{ij}	penalty term for a vertical heat match between i and j
q_{ij}	total heat exchanged between source i and sink j

q_i^{CU}	heat exchanged between hot stream i and cooling utility
q_j^{HU}	heat exchanged between heating utility and cold stream j
q_{ij}^{E}	total heat exchanged between source i and sink j
q_{ij}^{V}	maximum vertical heat transfer in a heat match between i and j
Q_m^{HU}	heat load of hot utility m
Q_n^{CU}	heat load of cold utility n
Q_{mk}^{HU}	heat load of hot utility m in interval k
Q_{nk}^{CU}	heat load of cold utility n in interval k
Q_{ijk}	heat exchanged between source i and sink j in interval or stage k
Q_{mjk}	heat exchanged between heating utility m and cold stream j in interval k
Q_{ink}	heat exchanged between hot stream i and cooling utility n in interval k
R_k	residual for temperature interval k
R_{ik}	residual for hot stream/heat source i for interval k
R_{mk}	residual for heating utility m for interval k
$s(a)$	size of element a in 3-partition problem
s_i	sum of heat supplies for heat source i
S_i	size of Knapsack item I
S_{ik}	heat supply for hot process stream i in interval k
$t_{i,k}^{\mathrm{H}}$	temperature of hot stream i at the end of stage k
$t_{j,k}^{\mathrm{C}}$	temperature of cold stream j at the end of stage k
$t_{k'}^{\mathrm{I},k}$	inlet temperature of stream k to the exchanger for k and k'
$t_{k'}^{\mathrm{O},k}$	outlet temperature of stream k form the exchanger for k and k'
T_k	inlet temperature of stream k
$T_i^{\mathrm{IN,H}}$	inlet temperature of hot stream I
$T_j^{\mathrm{IN,C}}$	inlet temperature of cold stream j
$T^{\mathrm{IN,CU}}$	inlet temperature of the cooling utility
$T^{\mathrm{IN,HU}}$	inlet temperature of the heating utility
$T_i^{\mathrm{OUT,H}}$	outlet temperature of hot stream i
$T_j^{\mathrm{OUT,C}}$	outlet temperature of cold stream j
$\Delta T_{ij}^{\max}$	maximum possible temperature drop through exchanger ij
U_{ij}	upper bound on heat exchange between source i and sink j
$\mathscr{U}_{ij}$	overall heat transfer coefficient for source i and sink j
y_{ij}	0–1 variable for heat match between source i and sink j
Y_i	0–1 variable for Knapsack item i
z_i^{CU}	0–1 variable for heat exchanger for hot stream i and the cooling utility
z_j^{HU}	0–1 variable for heat exchanger for the heating utility and cold stream j
z_{ijk}	0–1 variable for heat exchanger for hot stream i and the cold stream j in stage k
β	an upper bound on the number of heat matches
β_{ij}	exponent for area cost in exchanger ij
γ	weight factor for film heat transfer coefficient penalties
δ_{jk}	demand of heat at each heat sink j in interval k
$\delta_{jl}^{\mathscr{E}}$	heat content of sink j in enthalpy interval l
ϵ	weight factor for vertical heat transfer penalties
σ_{ik}	supply of heat at each heat source i in interval k
$\sigma_{ik}^{\mathscr{E}}$	heat content of source i in enthalpy interval l
ω	upper bound on heat exchange

Sets

A	set in the 3-partition problem
A_i	disjoint partitioned sets in the 3-partition problem
C	cold process streams
C_k	cold process streams present in interval k
CU	cooling utilities

CU_k	cooling utilities present in interval k
$\mathscr{E}$	enthalpy intervals
H	hot process streams
H_k	hot process streams present in interval k
H'_k	hot process streams present in intervals less than or equal to k
HCT	total stream set $\mathrm{HCT} = \mathscr{I} \cup \mathscr{J}$
HU	heating utilities
HU_k	heating utilities present in interval k
HU'_k	heating utilities present in intervals less than or equal to k
$\mathscr{I}$	heat sources $\mathscr{I} = H \cup \mathrm{HU}$
$\mathscr{I}_k$	heat sources present in interval k $\mathscr{I}_k = H_k \cup \mathrm{HU}_k$
$\mathscr{I}'_k$	heat sources present in intervals less than or equal to k $\mathscr{I}'_k = H'_k \cup \mathrm{HU}'_k$
$\mathscr{J}$	heat sinks $\mathscr{J} = C \cup \mathrm{CU}$
$\mathscr{J}_k$	heat sinks present in interval k $\mathscr{J}_k = C_k \cup \mathrm{CU}_k$
$\mathscr{NP}$	decision problems solved with nondeterministic polynomial time algorithms
$\mathscr{P}$	decision problems solved with deterministic polynomial time algorithms
$\mathscr{T}$	temperature intervals $\mathscr{T} = \{1,2,\ldots,K\}$
Z^+	positive integers

1. Introduction

Heat exchanger network synthesis (HENS) is one of the most extensively studied problems in chemical process synthesis. Its significance can be attributed to its role in controlling the costs of energy for a process. The basic HENS problem is the following:

Given:

- a set of hot process streams, each to be cooled from its supply temperature to its target temperature;
- a set of cold process streams, each to be heated from its supply temperature to its target temperature;
- the heat capacities and flow rates of the hot and cold process streams;
- the utilities available and the temperature or temperature range and costs for these utilities;

develop a heat exchanger network with the minimum annualized investment and annual operating costs.

The two primary methods for HENS are sequential and simultaneous synthesis methods. Sequential synthesis methods involve partitioning the basic problem according to its temperature range and then decomposing it further into various target subproblems, each solved sequentially subject to the solution of the prior target. On the other hand, simultaneous synthesis methods are concerned with solving the basic HENS problem with little or no decomposition into target subproblems.

Gundersen and Naess (1988) and Ježowski (1994a,b) have contributed thorough reviews on HENS. Furman and Sahinidis (in press) report that over 400 papers have been published on the subject over the last four decades. While HENS has received so much attention by the process systems engineering community, there have been no studies of its computational complexity and it is not known whether efficient (polynomial) algorithms might exist for the problem. More generally, while a variety of models, algorithms, and solution approaches have been developed for several types of process synthesis subproblems, not much is available in terms of formal computational complexity characterizations of these problems. A recent paper by Ahmed and Sahinidis (2000) is a notable exception in which the authors prove that the related multiperiod chemical process design and planning problem is $\mathscr{NP}$-hard. In the context of process control, Braatz and Russell (1999) have recently shown that the computation of robustness margins for large-scale systems is $\mathscr{NP}$-hard.

The purpose of this paper is to provide a formal computational complexity characterization of HENS. Section 2 serves as a brief review of computational complexity theory. In Section 3 the computational complexity of the target problems in the sequential synthesis of heat exchanger networks is studied. This section presents our main result, namely that the matches problem studied by Papoulias and Grossmann (1983) and Cerda and Westerberg (1983), as well as the vertical heat transfer problems of Gundersen and Grossmann (1990) and Gundersen, Duvold, and Hashemi-Ahmady (1996), are all $\mathscr{NP}$-hard in the strong sense. We then build on this result to show that more complex versions of HENS are also $\mathscr{NP}$-hard in the strong sense. In particular, Section 4 provides a complexity analysis of the existing HENS formulations for simultaneous synthesis. In Section 5 our complexity results are summarized and the implications of these results with regard to the general process synthesis problems are discussed. Finally, the Appendix A provides an alternate $\mathscr{NP}$-hardness proof via restriction of HENS to the Knapsack problem.

2. Computational complexity theory

This section is intended as an introduction to computational complexity theory and related topics. For a more in-depth discussion of the topic, the reader is referred to Garey and Johnson (1979), Papadimitriou and Steiglitz (1982), and Lewis and Papadimitriou (1998).

2.1. Basic terms

A *problem* is defined as a general question to be answered possessing parameters and free variables with unspecified values. A *problem description* includes a general listing of all its parameters and a statement of what properties and constraints the solution must satisfy. A *problem instance* includes a specification of particular values for all of the parameters in the problem. A *decision problem* is defined as a problem whose solution consists of an answer of either "yes" or "no". An *optimization problem* has the goal of finding a minimum or maximum value to a specified objective function subject to the constraints of the problem. To illustrate these terms, consider the decision problem of solving a system of linear equations. The problem description is as follows:

Given a system of linear algebraic equations with the form $Ax = b$, where A is an $m \times n$ matrix, b is a given m-vector, and x is the unknown solution n-vector to be determined, can the vector b be expressed as a linear combination of the columns of the matrix A?

An instance of this decision problem would include a specification of matrix A and vector b, i.e., $I = (A,b)$. On the other hand, an example of an optimization problem is the linear programming (LP) problem with cost coefficient vector c:

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

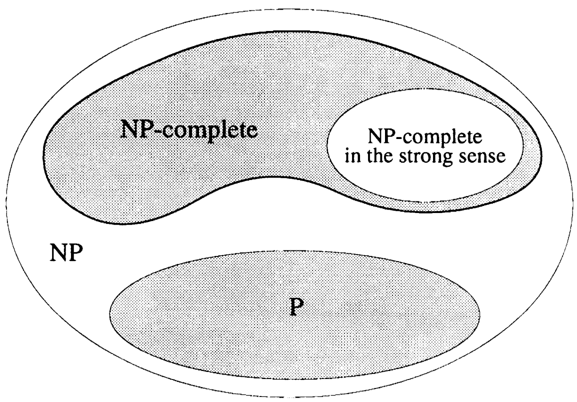

Fig. 1. The set $\mathcal{NP}$ and subsets.

An important trait of any given decision or optimization problem is the amount of time and number of calculations required to solve it. An *algorithm* is defined as a step-by-step procedure for solving a problem. A *polynomial time algorithm* is defined as an algorithm whose run time is polynomially bounded in the size of input and the logarithm of the size of the input values. Any algorithm whose time complexity cannot be bounded by a polynomial function of the problem input is considered an *exponential* time algorithm. *Pseudo-polynomial time algorithms* have run times polynomially bounded in the size of the input and largest input value and will be discussed in further detail in Section 2.7. Taking the example above, one algorithm for solving a system of n linear equations in n unknowns is Gaussian elimination. This algorithm has a run time $O(n^{2.376})$ and is thus polynomial with respect to n.

2.2. Complexity classes $\mathcal{P}$ and $\mathcal{NP}$

An important concept in complexity theory is the definition of the classes $\mathcal{P}$ and $\mathcal{NP}$. A decision problem is said to belong to class $\mathcal{P}$ if there is a deterministic algorithm that solves it in polynomial time. Class $\mathcal{NP}$ is the set of decision problems which can be solved with a nondeterministic polynomial time algorithm. A *deterministic algorithm* is another term for a sequential algorithm as it is normally used. A *nondeterministic algorithm* consists of two stages, an imaginary guessing stage where some possible solution is produced, and a deterministic checking stage in which it is determined whether or not the guessing stage produced a valid solution. A nondeterministic algorithm operates in "polynomial time" if, for some guess of the solution for a problem instance, the validity of the guess can be checked in polynomial time. Informally, the class $\mathcal{NP}$ is defined as the set of decision problems whose solution can be verified in polynomial time to determine whether or not it is valid.

The relationship between classes $\mathcal{P}$ and $\mathcal{NP}$ is a fundamental issue in computational complexity theory. It is obvious that class $\mathcal{P}$ is a subset of class $\mathcal{NP}$ as illustrated in Fig. 1. Every decision problem solvable by a polynomial time deterministic algorithm is solvable by a polynomial time nondeterministic algorithm as well. This is observed by ignoring the guessing stage and using a polynomial time deterministic algorithm as the verification stage of a nondeterministic algorithm. The most important issue in complexity theory is whether or not $\mathcal{P} = \mathcal{NP}$. No methods of converting polynomial time nondeterministic algorithms into polynomial time deterministic algorithms are known to exist. The ability of a nondeterministic algorithm to check an exponential number of solution possibilities in polynomial time is considered an unlikely prospect. For these reasons, although it has not been proven, it is generally assumed that $\mathscr{P} \neq \mathscr{N}\mathscr{P}$. Then the region $\mathscr{N}\mathscr{P} \backslash \mathscr{P}$ becomes very meaningful when characterizing the computational complexity of a decision problem.

2.3. Polynomial reductions and $\mathscr{N}\mathscr{P}$-completeness

$\mathscr{N}\mathscr{P}$-complete problems are those problems within set $\mathscr{N}\mathscr{P}$ for which no polynomial time algorithms exist, assuming $\mathscr{P} \neq \mathscr{N}\mathscr{P}$. A given problem is defined to be $\mathscr{N}\mathscr{P}$-complete if it is in $\mathscr{N}\mathscr{P}$ and any other $\mathscr{N}\mathscr{P}$-complete problem can be polynomially reduced (transformed) to it. Thus, if any $\mathscr{N}\mathscr{P}$-complete problem is solvable with a polynomial time algorithm, then *all* $\mathscr{N}\mathscr{P}$-complete problems are solvable in polynomial time. Conversely, if any problem in set $\mathscr{N}\mathscr{P}$ is found to be computationally intractable, then *all* $\mathscr{N}\mathscr{P}$-complete problems are intractable as well. From the definition, it is obvious that in order to prove that some problem π is $\mathscr{N}\mathscr{P}$-complete, it must be demonstrated that π is in the set $\mathscr{N}\mathscr{P}$, and that some known $\mathscr{N}\mathscr{P}$-complete problem can be polynomially reduced to π. The required initial $\mathscr{N}\mathscr{P}$-complete problem, SATISFIABILITY, is provided by Cook's Theorem (Cook, 1971).

A *polynomial reduction* (or transformation) from problem π_1 to problem π_2 exists if for any input data of π_1 the input data for π_2 can be constructed in polynomial time so that, once π_2 is solved, the solution to π_1 can subsequently be obtained in polynomial time. Informally, a reduction of π_1 to π_2 implies that π_2 may be considered as a special case of π_1, such that π_2 is at least as hard is π_1. A polynomial reduction from π_1 to π_2 is denoted by the expression $\pi_1 \propto \pi_2$. A *polynomial equivalence* between problems π_1 and π_2 is said to exist if $\pi_1 \propto \pi_2$ and $\pi_2 \propto \pi_1$. Problems in class $\mathscr{N}\mathscr{P}$-complete are then said to belong to the same equivalence class.

In general, there are three common techniques for proving $\mathscr{N}\mathscr{P}$-completeness: restriction, local replacement, and component design, which are explained in great detail in the text of Garey and Johnson (1979). A proof by *restriction* is the simplest approach consisting of showing that some problem π_1 contains some known $\mathscr{N}\mathscr{P}$-complete problem π_2 as a special case by placing restrictions on the instances of π_1. This method is frequently applied and is the primary method for the proofs of this paper.

2.4. Complexity: problems versus algorithms

The complexity of an algorithm is measured as a function of the size of the input and the size of the parameter values of the problem. For example, in the case of Gaussian elimination it was observed that the run time is $O(n^{2.376})$, a function of the size of the input matrix A. The computational complexity of a problem is characterized by the complexity of the most efficient algorithms for solving it. It has already been discussed that if a problem can be solved with a polynomial time algorithm, the problem belongs to class $\mathscr{P}$. The computational complexity of some problems can be determined without any information about algorithms for solving them. If it is shown that some problem π is $\mathscr{N}\mathscr{P}$-complete, then no polynomial time algorithm for solving π exists, under the assumption of $\mathscr{P} \neq \mathscr{N}\mathscr{P}$.

It is important to note that computational complexity analysis only considers the *worst case* solution times of the *best* possible algorithm over *all* possible instances. There are some cases in which the average solution time of an algorithm resembles polynomial time, even though its worst case bound is exponential. For example, in the case of LP, although purely polynomial time algorithms have been found by Khachian (1979) and Karmarkar (1984) and subsequently improved, for the most part the simplex algorithm performs better on average. However, the simplex algorithm is an exponential time algorithm in the worst case (Klee and Minty, 1972). Disregarding infrequent special cases of this type, it can be assumed that most problems determined to be computationally complex will not be solvable with algorithms that have polynomial time performance on average, even though they have exponential time bounds in the worst case.

2.5. Optimization problems and $\mathscr{N}\mathscr{P}$-hardness

Many problems whose complexity researchers may be interested fall outside of set $\mathscr{N}\mathscr{P}$. A different measure other than $\mathscr{N}\mathscr{P}$-completeness for the computational complexity for these types of problems is necessary. Any problem to which an $\mathscr{N}\mathscr{P}$-complete problem can be polynomially reduced, whether or not it is in set $\mathscr{N}\mathscr{P}$, is considered hard. A problem π is classified as $\mathscr{N}\mathscr{P}$-hard if an $\mathscr{N}\mathscr{P}$-complete problem can be polynomially reduced to π. $\mathscr{N}\mathscr{P}$-hard problems are characterized as being at least as hard as the $\mathscr{N}\mathscr{P}$-complete problems. In most cases, for an optimization problem it is unknown if it is in $\mathscr{N}\mathscr{P}$. Since the question of whether or not a problem is known to be in $\mathscr{N}\mathscr{P}$ is irrelevant when proving $\mathscr{N}\mathscr{P}$-hardness, a computational complexity characterization may be described for optimization problems.

2.6. Should not all mixed integer linear programming problems be $\mathscr{N}\mathscr{P}$-hard?

The complexity of a general type of optimization problem is not always readily apparent upon inspection. To some, it may seem "obvious" that all mixed integer linear programming (MILP) problems would be $\mathscr{N}\mathscr{P}$-hard. This is not the case, however. In some cases, MILP problems can be easily solved, but this is not readily apparent from their formulation. For the purpose of illustrating this idea, the example of the lot-sizing problem is useful (Nemhauser and Wolsey, 1988).

Lot-sizing problem: Consider the problem of satisfying demands d_t of a product over a number of time periods $t=1,\ldots,T$. If an amount of $y_t>0$ is produced in time period t, then the cost of $p_t y_t + c_t$ is incurred, where c_t is a fixed set-up charge independent of the amount produced, and p_t is a unit production cost. To avoid fixed charges, one may decide to produce large amounts in early time periods and use then to satisfy the demands of subsequent time periods. However, in this case an inventory holding cost of h_t per unit inventory is incurred in time period t. The objective is to minimize the total cost of production, set-up, and inventory storage while satisfying the demands.

The most intuitive formulation of the problem is obtained by defining y_t and s_t as the real variables for production and end storage, and designating the binary variable x_t to indicate whether or not $y_t>0$. This formulation is as follows:

$$\min \sum_{t=1}^{T} (p_t y_t + h_t s_t + c_t x_t)$$

$$s_{t-1} + y_t = d_t + s_t, \quad t=1,\ldots,T$$

$$y_t \leqslant \omega x_t, \quad t=1,\ldots,T$$

$$s_t \geqslant 0, \quad t=1,\ldots,T,$$

$$s_1 = s_T = 0$$

$$y_t \geqslant 0, \quad t=1,\ldots,T,$$

$$x_t \in \{0,1\}, \quad t=1,\ldots,T,$$

This problem can be alternately formulated by defining a real variable q_{it} as the quantity produced in period i to satisfy the demand in period $t \geqslant i$, and also defining x_t as the same as above. Another formulation, which is generally considered better, is as follows:

$$\min \sum_{t=1}^{T} \sum_{i=1}^{t} \left(p_i + \sum_{j=i}^{t-1} h_j \right) q_{it} + \sum_{t=1}^{T} c_t x_t$$

$$\sum_{i=1}^{t} q_{it} = d_t, \quad t=1,\ldots,T$$

$$q_{it} \leqslant d_t x_i, \quad i=1,\ldots,T, \; t=i,\ldots,T,$$

$$q_{it} \geqslant 0, \quad i=1,\ldots,T, \; t=i,\ldots,T,$$

$$x_t \in \{0,1\}, \quad t=1,\ldots,T.$$

When the integer constraints in the second formulation are relaxed, the resulting solution of the LP relaxation has $x_t \in \{0,1\}$ for all t values. Therefore, the lot-sizing problem can be solved in polynomial time using LP techniques.

This example illustrates that, in the case of MILP optimization problems, looks can sometimes be deceiving. In general, an (MILP) optimization formulation is merely a mathematical representation of a problem. Computational complexity, on the other hand, deals with the worst case behavior of the *best possible* mathematical treatment of the problem. Having a computational complexity characterization for a problem removes any doubts as to whether or not a technique may ever be found that terminates inside a polynomial time boundary. More specifically, assuming $\mathscr{P} \neq \mathscr{NP}$, $\pi \in \mathscr{NP}$ then implies that there is no polynomial-sized LP representation of problem π.

2.7. Pseudo-polynomial algorithms and strong $\mathscr{NP}$-completeness

For some instance I of a problem π, let the size of the input be denoted by N, and the size of the largest parameter value be denoted by B. Previously, polynomial time algorithms were defined as algorithms with run times polynomially bounded in the input of the problem. More formally, a polynomial time algorithm has a run time bounded polynomially in N and $\log B$. A *pseudo-polynomial time algorithm* has a run time polynomially bounded in N and B.

If, for every instance of a problem, B is polynomially bounded in N, then the existence of a pseudo-polynomial time algorithm implies the existence of a polynomial time algorithm. In general, however, the existence of a polynomial time algorithm does not imply the existence of one that runs in pseudo-polynomial time as an algorithm polynomial in B is exponential in $\log B$. Those $\mathscr{NP}$-complete problems for which no pseudo-polynomial time algorithm exists are said to be *strongly* $\mathscr{NP}$-complete. A similar definition exists for problems that are $\mathscr{NP}$-hard *in the strong sense*.

To prove that a given problem is $\mathscr{NP}$-complete in the strong sense, not only must it be shown to be in $\mathscr{NP}$ and reducible to another $\mathscr{NP}$-complete problem, but reducible to a *strongly* $\mathscr{NP}$-complete problem. It must also be noted that the size of input values must remain polynomially bounded throughout the reduction. Just as regular $\mathscr{NP}$-completeness proofs required an initial problem SATISFIABILITY for the reduction, the problem 3-PARTITION (Garey and Johnson, 1979) plays the same role in strong $\mathscr{NP}$-completeness proofs.

As an illustration of some of these ideas, consider the Knapsack problem:

Knapsack problem: Given a set of N items, profits p_i $(i=1,\ldots,N)$, sizes s_i $(i=1,\ldots,N)$, and a Knapsack capacity B_K, find a subset of the items with maximum total profit such that their total size is at most B_K.

The Knapsack problem is well known to be $\mathscr{NP}$-hard (Papadimitriou and Steiglitz, 1982). However, the proof utilizes very large numbers and leaves open the possibility of a pseudo-polynomial time algorithm. In fact, a pseudo-polynomial time algorithm is known to exist

(Ibarra and Kim, 1975) with $O(N^3P \log SB_K)$ run time, where $S = \max_i s_i$ and $P = \max_i p_i$. Therefore, all instance sets in which P is bounded polynomially by N allow this algorithm to solve the Knapsack problem within polynomial time. Further, if an input instance is constructed such that $p'_i = \lfloor p_i/D \rfloor$ where $D = P/((1/\epsilon)+1)n$, then the algorithm solves it in run time $O((N^3 \log SB_K)/\epsilon)$ approximating the exact solution within a relative performance guarantee of $1+\epsilon$.

2.8. Complexity issues for heat exchanger network synthesis

The above discussion touches on computational complexity issues of significance to HENS and process synthesis problems in general. The main conclusions are as follows:

(1) Computational complexity characterizations pertain to the worst case behavior of the best possible algorithm for a problem. As such, these characterizations are independent of the optimization formulation used to describe the problem (Section 2.6).

(2) To prove that a synthesis problem is "hard" it suffices to restrict it to a known hard problem (Section 2.3).

(3) For practical purposes, all parameters in a process synthesis problem are bounded numbers. It is precisely for this reason that $\mathcal{NP}$-hardness in the strong sense is the only meaningful hardness characterization. Simply showing that a problem is $\mathcal{NP}$-hard leaves open the possible existence of a pseudo-polynomial time algorithm (Section 2.7). Due to the boundedness of problem parameters, this will not rule out the possibility of a polynomial time algorithm.

We now proceed to characterize the computational complexity of subproblems that arise in the sequential synthesis of heat exchanger networks.

3. Sequential synthesis

One of the more prevalent methods for approaching the HENS problem is the sequential synthesis method. This method involves partitioning the problem, usually by partitioning the temperature range of the problem into temperature intervals and possibly subnetworks, in accordance with a set of rules governing one of the pinch or pseudo-pinch design methods. The basic HENS problem is decomposed into three subproblems: the minimum utilities cost, the minimum number of matches, and the minimum cost network problems. These problems are then solved according to the heuristic of finding the minimum cost network subject to the minimum number of units which is subject to the minimum utilities cost (Biegler, Grossmann, and Westerberg, 1997):

min Network area cost
s.t. Minimum number of units
  s.t. Minimum utilities cost.

Mathematical programming-based sequential synthesis requires the following procedure:

1. Partition the problem into temperature intervals based on a minimum approach temperature (Linnhoff and Flower, 1978; Cerda, Westerberg, Mason, and Linnhoff, 1983; Biegler, Grossmann, and Westerberg, 1997) and sometimes enthalpy intervals as well(Gundersen and Grossmann, 1990; Gundersen, Duvold, and Hashemi-Ahmady, 1996). Also, the problem is sometimes divided into subnetworks based on the pinch point(s).
2. Solve for the minimum utilities cost. The problem formulations of Papoulias and Grossmann (1983) and Cerda, Westerberg, Mason, and Linnhoff (1983) are the most commonly used.
3. Solve for the minimum number of heat exchange matches and the heat load distribution based on the utility data of the previous step. The formulations of Papoulias and Grossmann (1983) and Cerda and Westerberg (1983) are the most common, while the vertical heat transfer formulations of Gundersen and Grossmann (1990) and Gundersen, Duvold, and Hashemi-Ahmady (1996) are also used.
4. Derive a minimum cost network based on the matches and heat load information of the two previous steps. The superstructure-based formulation of Floudas, Ciric, and Grossmann (1986) is typically used.

There have been various implementations of sequential synthesis (Floudas, Ciric, and Grossmann, 1986; Saboo, Morari, and Colberg, 1986), all of which maintain the basic sequence outlined above. We now characterize the computational complexity of the problems that need to be solved in the various stages of the sequential approach.

3.1. Utilities problems

Given

- sets H of hot process streams and C of cold process streams;
- sets HU of heating utilities and CU of cooling utilities;
- a set of temperature intervals $\mathcal{T} = \{1,\ldots,K\}$ with decreasing temperatures as interval number increases;
- sets H_k of hot streams and HU_k of heating utilities present in interval k;
- sets C_k of cold streams and CU_k of cooling utilities present in interval k;
- a supply S_{ik} of heat for each hot process stream $i \in H$ in interval k;

- a demand D_{jk} of heat for each cold process stream $j \in C$ in interval k;
- costs c_m and c_n for utilities where $m \in \mathrm{HU}$ and $n \in \mathrm{CU}$;

the minimum utilities cost problem consists of determining the lowest cost amount of utilities required in the heat exchanger network being designed.

Several problem formulations exist for determining the minimum utilities cost for a heat exchanger network. Papoulias and Grossmann (1983) proposed two formulations based on a transshipment model. Their most simple formulation includes energy balances for the temperature intervals and does not allow for constraints on the matches to be taken into account:

Problem UU

$$\min \sum_{m \in \mathrm{HU}} c_m Q_m^{\mathrm{HU}} + \sum_{n \in \mathrm{CU}} c_n Q_n^{\mathrm{CU}}$$

subject to

$$R_k - R_{k-1} - \sum_{m \in \mathrm{HU}_k} Q_m^{\mathrm{HU}} + \sum_{n \in \mathrm{CU}_k} Q_n^{\mathrm{CU}} = \sum_{i \in H_k} S_{ik} - \sum_{j \in C_k} D_{jk}, \; k \in \mathcal{T},$$

$$Q_m^{\mathrm{HU}} \geqslant 0, \; m \in \mathrm{HU},$$

$$Q_n^{\mathrm{CU}} \geqslant 0, \; n \in \mathrm{CU},$$

$$R_k \geqslant 0, \; k \in \mathcal{T},$$

$$R_0 = R_K = 0.$$

The other formulation of Papoulias and Grossmann (1983), which we will denote as problem U, allows for constraints regarding forbidden and required matches. Cerda, Westerberg, Mason, and Linnhoff (1983) proposed a minimum utilities cost problem formulation, denoted here as model U', based on a transportation problem model. It is obvious from observation that target problems UU, U, and U' are all minimum cost network flow problems the size of which is polynomial in the number of process and utility streams. Ahuja, Magnanti, and Orlin, (1993) review many strongly polynomial algorithms that can solve such problems in no more than $\mathrm{O}(\bar{m} \log \bar{n}(\bar{m} + \bar{n} \log \bar{n}))$, where $\bar{m}$ and $\bar{n}$ are the numbers of arcs and nodes of the underlying graph of the network. Thus, the minimum utilities cost problems are solvable in polynomial time or are "easy".

Theorem 1. *The minimum utilities cost problem is in* $\mathcal{P}$.

Another way to characterize complexity of the minimum utilities problem is by recognizing that formulations UU, U, and U' are all linear programs (LPs) and can therefore be solved in polynomial time. In particular, Khachian (1979) developed the first polynomial time algorithm for solving LP problems in general, leading to the conclusion that LP problems are in $\mathcal{P}$. However, algorithms for general LPs exhibit much worse complexity than the above-mentioned algorithms for minimum cost network flow problems.

While it was straightforward to characterize the computational complexity of the minimum utilities problems, we next present a much more profound result regarding the complexity of the matches problem.

3.2. The minimum number of matches problem

Given

- a set of heat sources (hot streams and heating utilities) $\mathcal{I} = \{1, \ldots ,m\}$;
- a set of heat sinks (cold streams and cooling utilities) $\mathcal{J} = \{1, \ldots ,n\}$;
- a set of temperature intervals $\mathcal{T} = \{1, \ldots , K\}$ with decreasing temperatures as interval number increases;
- sets $\mathcal{I}'_k$ of hot streams and heating utilities present in intervals less than or equal to k;
- sets $\mathcal{J}_k$ of cold streams and cooling utilities present in interval k;
- a supply σ_{ik} of heat at each source $i \in \mathcal{I}$ in interval k;
- a demand δ_{jk} for heat at each sink $j \in \mathcal{J}$ in interval k;
- a cost c_{ij} for each possible match of sinks and sources that indicates a preference for particular matches;

the minimum number of matches problem consists of determining the minimum cost set of matches satisfying the supplies and demands of heat for a heat exchanger network or subnetwork. A match is defined as an exchange of heat between a heat source and cold source determined by the solution of the given problem. We consider the formulation of Papoulias and Grossmann (1983):

Problem M

$$\min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} y_{ij}$$

subject to

$$R_{ik} - R_{ik-1} + \sum_{j \in \mathcal{J}_k} Q_{ijk} = \sigma_{ik}, \quad i \in \mathcal{I}'_k, k \in \mathcal{T},$$

$$\sum_{i \in \mathcal{I}'_k} Q_{ijk} = \delta_{jk}, \quad j \in \mathcal{J}_k, \; k \in \mathcal{T},$$

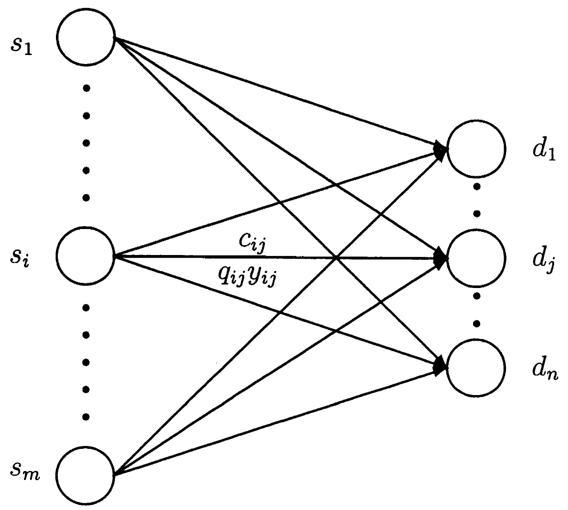

Fig. 2. Problem $M1$.

$$\sum_{k=1}^{K} Q_{ijk} - U_{ij}y_{ij} \leqslant 0, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

$$R_{ik} \geqslant 0, \quad i\in\mathscr{I},$$

$$R_{i0} = R_{iK} = 0, \quad i\in\mathscr{I},$$

$$Q_{ijk} \geqslant 0, \quad i\in\mathscr{I},\ j\in\mathscr{J},\ k\in\mathscr{T},$$

$$y_{ij}\in\{0, 1\}, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

where $U_{ij} = \min\{s_i, d_j\}$, with $s_i := \Sigma_{k=1}^{K}\sigma_{ik}$ and $d_j := \Sigma_{k=1}^{K}\delta_{jk}$. Then, the feasibility requirement

$$\sum_{i\in\mathscr{I}} s_i = \sum_{j\in\mathscr{J}} d_j \tag{1}$$

for problem M is guaranteed by the solution of the minimum utilities cost problem prior to solving the matches problem in sequential HENS. The first two constraints of problem M represent energy balances, while the third is a logical constraint. Another matches problem M' used by Cerda and Westerberg (1983) is equivalent to M.

When problem M is restricted to the special case which has only one temperature interval ($K = 1$), the residual heat variables (R_{ik}) are cancelled and the k indices can be removed. This case is presented graphically in Fig. 2 and can be formulated as follows:

Problem M1

$$\min \sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} c_{ij}y_{ij}$$

subject to

$$\sum_{j\in\mathscr{J}} q_{ij} = s_i, \quad i\in\mathscr{I},$$

$$\sum_{i\in\mathscr{I}} q_{ij} = d_j, \quad j\in\mathscr{J},$$

$$q_{ij} \leqslant U_{ij}y_{ij}, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

$$q_{ij} \geqslant 0, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

$$y_{ij}\in\{0, 1\}, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

When solving a minimum number of matches problem, often there will be no preference on matches and all the cost coefficients in the objective function are unity. First consider the restriction of problem M to the special case $M1$. Then further restrict $M1$ subject to the following criteria:

$$c_{ij} = 1, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

$$|\mathscr{I}| = 3n,$$

$$|\mathscr{J}| = n,$$

$$d_j = B, \quad j\in\mathscr{J},$$

$$\frac{B}{4} < s_i < \frac{B}{2}, \quad i\in\mathscr{I},$$

This choice of parameters clearly implies $U_{ij} = s_i$ for $i\in\mathscr{I}$, $j\in\mathscr{J}$. Now the resulting problem formulation can be stated as the following:

Problem M1-1

$$\min \sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} y_{ij}$$

subject to

$$\sum_{j\in\mathscr{J}} q_{ij} = s_i, \quad i\in\mathscr{I},$$

$$\sum_{i\in\mathscr{I}} q_{ij} = B, \quad j\in\mathscr{J},$$

$$q_{ij} \leqslant s_i y_{ij}, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

$$q_{ij} \geqslant 0, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

$$y_{ij}\in\{0, 1\}, \quad i\in\mathscr{I},\ j\in\mathscr{J}.$$

This instance arises in a minimum number of matches problem with only one temperature interval in which all of the cold streams require the same amount of heat energy. Also, there is no cost difference for heat exchangers for different matches. Thus, the cost coefficients are all equal and, therefore, can be set to unity without loss of generality.

Problem $M1$-1 is at least as hard as its corresponding decision problem stated below.

Problem M1-1D

Instance: Finite set $\mathscr{I}$ where $|\mathscr{I}| = 3n$, finite set $\mathscr{J}$ where $|\mathscr{J}| = n$, a number B, and a size s_i such that $(B/4) < s_i < (B/2)$ for $i\in\mathscr{I}$.
Question: Does there exist a feasible solution of matches $\{y_{ij} | i\in\mathscr{I}, j\in\mathscr{J}\}$ and heat loads $\{q_{ij} | i\in\mathscr{I}, j\in\mathscr{J}\}$ such that:

$$\sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} y_{ij} \leqslant \beta := 3n,$$

$$\sum_{j\in\mathscr{J}} q_{ij} = s_i, \quad i\in\mathscr{I},$$

$$\sum_{i\in\mathcal{I}} q_{ij} = B, \quad j\in\mathcal{J},$$

$$0 \leqslant q_{ij} \leqslant s_i y_{ij}, \quad i\in\mathcal{I},\ j\in\mathcal{J},$$

$$y_{ij}\in\{0, 1\}, \quad i\in\mathcal{I},\ j\in\mathcal{J}.$$

As there are $3n$ sources, it is required that $\beta \geqslant 3n$. With β set equal to $3n$, an equivalence with 3-partition may be shown. The 3-partition problem is given as follows (Garey and Johnson, 1979):

Problem 3PART

Instance: A set A consisting of $3n$ elements, a bound $B\in\mathscr{Z}^+$, for each $a\in A$ a size $s(a)\in\mathscr{Z}^+$ such that each $s(a)$ satisfies $(B/4) < s(a) < (B/2)$ and such that $\Sigma_{a\in A}s(a) = nB$.
Question: Can A be partitioned into n disjoint sets A_1, A_2, …, A_n such that $\Sigma_{a\in A_j}s(a) = B$ for $1 \leqslant j \leqslant n$?

Theorem 2. *Problem 3PART has a feasible solution if and only if problem M1-1D has a feasible solution.*

Proof 1. *Necessity*. Let $S = \{s(a)|a\in A\}$ and consider instance A, S, n of *3PART*. Construct an instance of M1-1D by setting $\mathcal{I} = A$ and $s_i = s(a)$ for $i = a$. A solution to *3PART* has n disjoint sets A_1, …, A_n each with three elements of A (Garey and Johnson, 1979). Let $\mathcal{J} = \{j|\{A_j\}$ is a solution 3PART$\}$. Using a bipartite graph representation $G(\mathcal{I}\cup\mathcal{J},E)$ of problem M1-1D with independent node sets $\mathcal{I}$ and $\mathcal{J}$, let each node j in $\mathcal{J}$ be incident to the elements of A that are in set A_j. Let $y_{ij} = 1$ if an arc between some $i\in\mathcal{I}$ and $j\in\mathcal{J}$ exists, and zero otherwise. Then $\Sigma_{i\in\mathcal{I}}\Sigma_{j\in\mathcal{J}}y_{ij} = 3n = \beta$ and each $i\in\mathcal{I}$ has exactly one arc to a node in $\mathcal{J}$. Then

$$\sum_{a\in A_i} s(a) = B \Rightarrow \sum_{i\in\mathcal{I}} s_i y_{ij} = B, \quad \forall j\in\mathcal{J}.$$

Let

$$q_{ij} = \begin{cases} s_i & \text{for } y_{ij} = 1, \\ 0 & \text{for } y_{ij} = 0. \end{cases}$$

Then $\Sigma_{i\in\mathcal{I}}s_i y_{ij} = \Sigma_{i\in\mathcal{I}}q_{ij} = B$ for $j\in\mathcal{J}$. Also since entire sources go to one destination node, $q_{ij} = s_i$ for $j\in\mathcal{J}$. Thus $\Sigma_{j\in\mathcal{J}}q_{ij} = s_i$ and $s_i y_{ij} = q_{ij}$ for $i\in\mathcal{I}, j\in\mathcal{J}$. Now all feasibility constraints of M1-1D have been met.

Sufficiency. Given a solution to some instance of M1-1D, construct an instance of 3PART by setting $A = \mathcal{I}$ and $s(a) = s_i$ for $a = i$. The value for β in M1-1D is $3n$ and in the bipartite graph representation of M1-1D at least one arc representing a match must extend from each node in $\mathcal{I}$. Then, $\Sigma_{i\in\mathcal{I}}\Sigma_{j\in\mathcal{J}}y_{ij} = 3n$. Since $(B/4) < s_i < (B/2)$, there are three arcs incident to each node $j\in\mathcal{J}$. Thus the n nodes in $\mathcal{J}$ each represent a disjoint partition of $G(\mathcal{I}\cup\mathcal{J},E)$ into components consisting of one element of $\mathcal{J}$ and three elements of $\mathcal{I}$. Set $\mathcal{I}$ can therefore be partitioned into m disjoint sets implying that A can be partitioned this way as well. Since only one arc extends from each node in $\mathcal{I}$, $q_{ij} = s_i$. Then the constraint $\Sigma_{i\in\mathcal{I}}q_{ij} = B$ for all $j\in\mathcal{J}$ of M1-1D satisfies the constraint $\Sigma_{a\in A_j}s(a) = B$ for all $A_j \subseteq A$.

It is well known that *3PART* is $\mathcal{NP}$-complete in the strong sense (Garey and Johnson, 1979). From the preceding equivalence relation, the following results:

Corollary 1. *Problem M1-1D is complete in the strong sense.*

Since problem M1-1D is $\mathcal{NP}$-complete in the strong sense, its corresponding optimization problem is at least as hard, and thus problem M1-1 is $\mathcal{NP}$-hard in the strong sense.

Corollary 2. *The minimum number of matches problem with unit cost coefficients (M1-1) is $\mathcal{NP}$-hard in the strong sense.*

Since M1-1 is a special case of problem M, the main result of this section follows:

Corollary 3. *The minimum number of matches problem (M) is $\mathcal{NP}$-hard in the strong sense.*

3.3. The vertical heat transfer minimum number of matches problem

Since the minimum number of matches problems may have more than one optimal solution exhibiting the same number of matches, a vertical heat transfer criterion for determining which solution is more desirable was proposed by Gundersen and Grossmann (1990). The goal is to determine the set of matches that will produce a network with lower capital costs due to exchanger area. In this model, not only is the temperature range of the problem partitioned into intervals ($\mathcal{T}$), but the enthalpy is also partitioned into intervals ($\mathcal{E}$):

Problem VM

$$\min \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} y_{ij} + \epsilon \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} P_{ij}$$

subject to

$$R_{ik} - R_{ik-1} + \sum_{j\in\mathcal{J}_k} Q_{ijk} = \sigma_{ik}, \quad i\in\mathcal{I}'_k,\ k\in\mathcal{T}, \tag{2}$$

$$\sum_{i\in\mathcal{I}'_k} Q_{ijk} = \delta_{jk}, \quad j\in\mathcal{J}_k, k\in\mathcal{T}, \tag{3}$$

$$q^E_{ij} = \sum_{k=1}^{K} Q_{ijk}, \quad i\in\mathcal{I}, j\in\mathcal{J}, \tag{4}$$

$$q^E_{ij} - U_{ij}y_{ij} \leqslant 0, \quad i\in\mathcal{I}, j\in\mathcal{J}, \tag{5}$$

$$L_{ij}y_{ij} - q^{E}_{ij} \leqslant 0, \quad i\in\mathscr{I}, j\in\mathscr{J}, \tag{6}$$

$$q^{E}_{ij} - P_{ij} \leqslant q^{V}_{ij}, \quad i\in\mathscr{I}, j\in\mathscr{J}, \tag{7}$$

$$R_{ik} \geqslant 0, \quad i\in\mathscr{I}, \tag{8}$$

$$R_{i0} = R_{iK} = 0, \quad i\in\mathscr{I}, \tag{9}$$

$$Q_{ijk} \geqslant 0, \quad i\in\mathscr{I}, j\in\mathscr{J}, k\in\mathscr{T}, \tag{10}$$

$$P_{ij} \geq 0, \quad i\in\mathscr{I}, j\in\mathscr{J}, \tag{11}$$

$$y_{ij}\in\{0,1\}, \quad i\in\mathscr{I}, j\in\mathscr{J}, \tag{12}$$

where $U_{ij} = \min\{s_i,d_j\}$. The basic feasibility condition provided by the solution of the minimum utilities cost problem prior to this problem remains Eq. (1). Parameter q^{V}_{ij} is the maximum vertical heat transfer in a match calculated from the heat contents of the streams in an enthalpy interval:

$$q^{V}_{ij} = \sum_{l\in\mathscr{E}} \min\{\sigma^{\mathscr{E}}_{il},\delta^{\mathscr{E}}_{jl}\}.$$

Problem VM was extended by Gundersen, Duvold, and Hashemi-Ahmady (1996) to penalize streams with poor film heat transfer coefficients:

Problem EVM

$$\min \alpha \sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} y_{ij} + \epsilon \sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} P_{ij} + \gamma \sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} y_{ij}\left[1 - \min\left(\frac{h_i}{h_j}, \frac{h_j}{h_i}\right)\right]$$

subject to (2) through (12).

Problem *EVM* has the new weight factor γ in the objective function which is usually used for normalizing the penalty over the maximum possible number of matches.

Let problems *VM* and *EVM* be restricted to the special case with only one temperature interval (i.e., $K = 1$) and with the following specific parameter values:

$$\epsilon = 0$$

$$\gamma = 0$$

$$L_{ij} = 0, \quad i\in\mathscr{I}, j\in\mathscr{J}.$$

The resulting problem has the following formulation:

Problem VM1

$$\min \sum_{i\in\mathscr{I}} \sum_{j\in\mathscr{J}} y_{ij}$$

subject to

$$\sum_{j\in\mathscr{J}} q^{E}_{ij} = s_i, \quad i\in\mathscr{I},$$

$$\sum_{i\in\mathscr{I}} q^{E}_{ij} = d_j, \quad j\in\mathscr{J},$$

$$q^{E}_{ij} \geqslant 0, \quad i\in\mathscr{I}, j\in\mathscr{J},$$

and constraints (5), (7), (11), and (12).

Since variables P_{ij} no longer appear in the objective function, constraints (7) and (11) are completely redundant and can therefore be removed without altering the feasibility or solution of *VM1*. The resulting equivalent formulation is nearly the same as problem M1-1 except that in M1-1 parameter $d_j = B$ for all $j\in\mathscr{J}$. It is obvious by this observation that M1-1 is a special case of *VM1*. Thus the implication is that M1-1 is a special case of *VM* and *EVM*.

Corollary 4. *The vertical heat transfer minimum number of matches problem VM and its extended version EVM are $\mathscr{N}\mathscr{P}$-hard in the strong sense.*

4. Simultaneous synthesis

Simultaneous synthesis of networks involves solving HENS with little or no decomposition of the problem as applied in sequential synthesis. These methods are primarily MINLP formulations of the HENS problem subject to several simplifying assumptions. We address a number of such problems. The basic approach is to show that they include the matches problem as a subproblem and are therefore at least as hard.

4.1. Simultaneous match-network problem

Given:

- a set of heat sources (hot streams and heating utilities) $\mathscr{I} = \{1,\ldots,m\}$;
- a set of heat sinks (cold streams and cooling utilities) $\mathscr{J} = \{1,\ldots,n\}$;
- combined stream set $\mathrm{HCT} = \mathscr{I} \cup \mathscr{J}$ with heat capacity flow rates F_k and inlet temperature T_k where $k\in\mathrm{HCT}$;
- a set of temperature intervals $\mathscr{T} = \{1,\ldots,K\}$ with decreasing temperatures as interval number increases;
- sets $\mathscr{I}'_t$ of hot streams and heating utilities present in intervals less than or equal to t;
- sets $\mathscr{T}_t$ of cold streams and cooling utilities present in interval t;
- a supply σ_{it} of heat at each source i in interval t;
- a demand δ_{jt} for heat at each sink j in interval t;
- cost parameters c_{ij} and β_{ij} for the exchangers;
- overall heat transfer coefficients $\mathscr{U}_{ij}$ for the exchangers,

the simultaneous match-network synthesis problem of Floudas and Ciric (1989) solves for the matches, heat load distribution, and network configuration all at the same time. It makes use of the superstructure illustrated in Fig. 3 and involves the following formulation:

Problem MN

$$\min \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}\left(\frac{q_{ij}}{\mathcal{U}_{ij}\mathrm{LMTD}_{ij}}\right)^{\beta_{ij}} y_{ij}$$

subject to

$$\left.\begin{array}{ll}
R_{it}-R_{it-1}+\sum_{j\in\mathcal{J}_t} Q_{ijt}=\sigma_{it} & i\in\mathcal{I}'_t,\ t\in\mathcal{T}\\
\sum_{i\in\mathcal{I}'_t} Q_{ijt}=\delta_{jt} & j\in\mathcal{J}_t,\ t\in\mathcal{T}\\
q_{ij}=\sum_{t=1}^{K} Q_{ijt}, & i\in\mathcal{I},\ j\in\mathcal{J}\\
q_{ij}=U_{ij}y_{ij}\leqslant 0 & i\in\mathcal{I},\ j\in\mathcal{J}\\
\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} y_{ij}\leqslant N_{\max} & \\
R_{it}\geqslant 0 & i\in\mathcal{I},\ t\in\mathcal{T}\\
R_{i0}=R_{iK}=0 & i\in\mathcal{I}\\
Q_{ijt}\geqslant 0 & i\in\mathcal{I},\ j\in\mathcal{J},\ t\in\mathcal{T}\\
y_{ij}\in\{0,1\} & i\in\mathcal{I},\ j\in\mathcal{J}
\end{array}\right\}\quad(13)$$

where $U_{ij}=\min\{s_i,d_j\}$.

The objective of this problem is the lowest annual cost for the heat exchanger network subject to the utilities parameters determined in the minimum utilities cost problem. This formulation requires that the HENS problem be partitioned into temperature intervals in the same manner as for sequential synthesis methods. Constraint set (13) represents the energy balances for heat matches. Constraints (14) are the flow balances and heat equations graphically represented in Fig. 3.

Theorem 3. *The simultaneous match-network problem MN is $\mathcal{NP}$-hard in the strong sense.*

Proof 2. When problem MN is restricted to the special case where $N_{\max}=|\mathcal{I}|+|\mathcal{J}|+1$, $\beta_{ij}=0$ for all combinations of i and j, and with only one temperature interval ($K=1$), the objective function becomes the following:

$$\min \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}y_{ij}.$$

$$\left.\begin{array}{rl}
\sum_{k'} f_{k'}^{I,k}=F_k & k\in HCT\\
f_{k'}^{I,k}+\sum_{k''} f_{k',k''}^{B,k}-f_{k'}^{E,k}=0 & k',k\in HCT\\
f_{k'}^{O,k}+\sum_{k''} f_{k'',k'}^{B,k}-f_{k'}^{E,k}=0 & k',k\in HCT\\
T_k f_{k'}^{I,k}+\sum_{k''} f_{k',k''}^{B,k}t_{k''}^{O,k}-f_{k'}^{E,k}t_{k'}^{I,k}=0 & k',k\in HCT\\
q_{ij}=f_j^{E,i}(t_j^{I,i}-t_j^{O,i}) & i\in\mathcal{I},j\in\mathcal{J}\\
q_{ij}=f_i^{E,j}(t_i^{O,j}-t_i^{I,j}) & i\in\mathcal{I},j\in\mathcal{J}\\
dt_{ij}^1=t_j^{I,i}-t_i^{O,j} & i\in\mathcal{I},j\in\mathcal{J}\\
dt_{ij}^2=t_j^{O,i}-t_i^{I,j} & i\in\mathcal{I},j\in\mathcal{J}\\
dt_{ij}^1\geq EMAT & i\in\mathcal{I},j\in\mathcal{J}\\
dt_{ij}^2\geq EMAT & i\in\mathcal{I},j\in\mathcal{J}\\
LMTD_{ij}=\frac{dt_{ij}^1-dt_{ij}^2}{\ln\frac{dt_{ij}^1}{dt_{ij}^2}} & i\in\mathcal{I},j\in\mathcal{J}\\
f_j^{I,i}\geq 0 & i\in\mathcal{I},j\in\mathcal{J}\\
f_i^{I,j}\geq 0 & i\in\mathcal{I},j\in\mathcal{J}\\
f_{j,k}^{B,i}\geq 0 & i\in\mathcal{I},j,k\in\mathcal{J}\\
f_{i,k}^{B,j}\geq 0 & i,k\in\mathcal{I},j\in\mathcal{J}\\
f_j^{O,i}\geq 0 & i\in\mathcal{I},j\in\mathcal{J}\\
f_i^{O,j}\geq 0 & i\in\mathcal{I},j\in\mathcal{J}\\
f_j^{E,i}-F_iy_{ij}\leq 0 & i\in\mathcal{I},j\in\mathcal{J}\\
f_i^{E,j}-F_jy_{ij}\leq 0 & i\in\mathcal{I},j\in\mathcal{J}\\
f_j^{E,i}\geq \frac{q_{ij}}{\Delta T_{ij}^{\max}} & i\in\mathcal{I},j\in\mathcal{J}\\
f_i^{E,j}\geq \frac{q_{ij}}{\Delta T_{ij}^{\max}} & i\in\mathcal{I},j\in\mathcal{J}
\end{array}\right\}\quad(14)$$

where $U_{ij}=\min\{s_i,d_j\}$.

This special case is only concerned with fixed costs of heat exchangers and will be called problem *MN1*. It should be noted that constraints (13) are the constraint set of problem *M*. Floudas, Ciric, and Grossmann (1986) proved that, in sequential synthesis for a set of matches feasible for the constraints of problem *M*, there must exist a feasible solution to the network derivation problem, which will be denoted as problem *N*. Since constraints (14) represent the same constraint set as problem *N* as reformulated by Floudas and Ciric (1989), any solution feasible for constraints (13) has a feasible solution to constraints (14). Thus, constraints (14) may be removed without altering the solution to the modified objective function for problem *MN1* in an equivalent formulation and thus problem *MN1* is equivalent to problem *M*1.

4.2. Heat exchanger network synthesis without decomposition

Given:

- sets H of hot process streams and C of cold process streams;
- combined stream set $HCT = \mathscr{I} \cup \mathscr{J}$ with heat capacity flow rates F_k and inlet temperature T_k where $k \in HCT$;
- a set of temperature intervals $\mathscr{T} = \{1,\ldots,K\}$ with decreasing temperatures as interval number increases;
- sets H'_t of hot streams and HU'_t of heating utilities present in intervals less than or equal to t;
- sets C_t of cold streams and CU_t of cooling utilities present in interval t;
- a supply S_{it} of heat for each hot process stream i in interval t;
- a demand D_{jt} of heat for each cold process stream j in interval t;
- cost parameters c_{ij} and β_{ij} for the exchangers and c_m and c_n for utilities;
- overall heat transfer coefficients for the exchangers $\mathscr{U}_{ij}$,

the HENS without decomposition formulation of Ciric and Floudas (1991) solves for a heat exchanger network with the lowest total annual cost subject to the temperature intervals partitioning based on the minimum approach temperature. This formulation requires that the HENS problem be partitioned into temperature intervals in the same manner as for sequential synthesis methods. The constraints are very similar to those of problem *MN* with the addition of consideration for utility streams:

Problem UMN

$$\min \sum_{i\in\mathscr{I}}\sum_{j\in\mathscr{J}} c_{ij}\left(\frac{q_{ij}}{\mathscr{U}_{ij}\mathrm{LMTD}_{ij}}\right)^{\beta_{ij}} y_{ij} + \sum_{m\in\mathrm{HU}} c_m Q_m^{\mathrm{HU}}$$

$$+ \sum_{n\in\mathrm{CU}} c_n Q_n^{\mathrm{CU}}$$

subject to

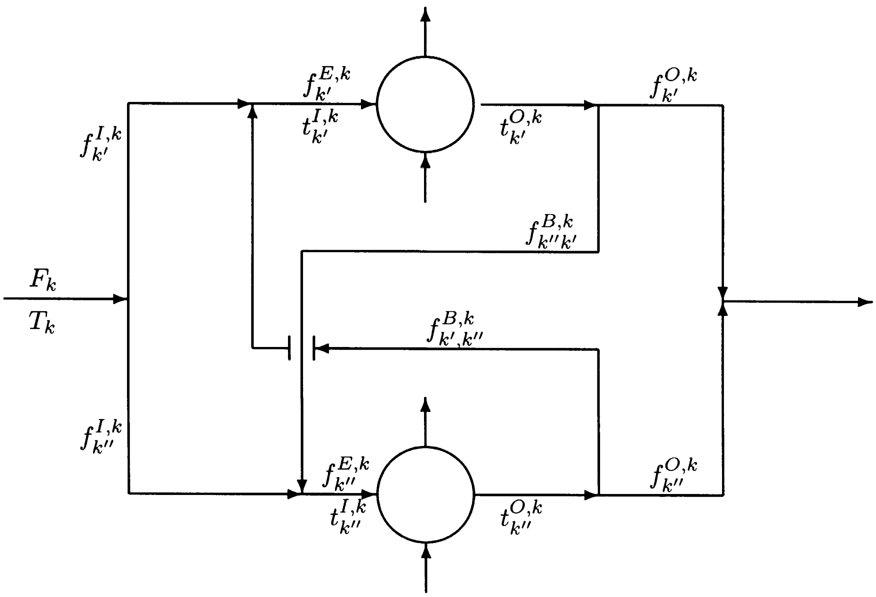

Fig. 3. Network superstructure.

Problem *UMN*

$$\min \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}\left(\frac{q_{ij}}{U_{ij}LMTD_{ij}}\right)^{\beta_{ij}} y_{ij} + \sum_{m\in HU} c_m Q_m^{HU} + \sum_{n\in CU} c_n Q_n^{CU}$$

subject to

$$\left.\begin{array}{rl}
R_{it} - R_{it-1} + \sum_{j\in C_t} Q_{ijt} + \sum_{n\in CU_t} Q_{int} = S_{it} & i\in H'_t, t\in\mathcal{T} \\
R_{mt} - R_{mt-1} + \sum_{j\in C_t} Q_{mjt} - Q_{mt}^{HU} = 0 & m\in HU'_t, t\in\mathcal{T} \\
\sum_{i\in H'_t} Q_{ijt} + \sum_{m\in HU'_t} Q_{mjt} = D_{jt} & j\in C_t, t\in\mathcal{T} \\
\sum_{i\in H'_t} Q_{int} - Q_{nt}^{CU} = 0 & n\in CU_t, t\in\mathcal{T} \\
q_{ij} = \sum_{t=1}^{K} Q_{ijt} & i\in\mathcal{I}, j\in\mathcal{J} \\
Q_m^{HU} = \sum_{t=1}^{K} Q_{mt}^{HU} & m\in HU \\
Q_n^{CU} = \sum_{t=1}^{K} Q_{nt}^{CU} & n\in CU \\
q_{ij} - U_{ij}y_{ij} \le 0 & i\in\mathcal{I}, j\in\mathcal{J} \\
q_{ij} - L_{ij}y_{ij} \ge 0 & i\in\mathcal{I}, j\in\mathcal{J} \\
\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} y_{ij} \le N_{\max} & \\
R_{it} \ge 0 & i\in\mathcal{I}, t\in\mathcal{T} \\
R_{i0} = R_{iK} = 0 & i\in\mathcal{I} \\
Q_{ijt} \ge 0 & i\in\mathcal{I}, j\in\mathcal{J}, t\in\mathcal{T} \\
Q_{mt}^{HU} \ge 0 & m\in HU, t\in\mathcal{T} \\
Q_{nt}^{CU} \ge 0 & n\in CU, t\in\mathcal{T} \\
y_{ij} \in \{0,1\} & i\in\mathcal{I}, j\in\mathcal{J}
\end{array}\right\}$$

$$\sum_{k'} f_{k'}^{\mathrm{I},k} = F_k, \quad k \in HCT,$$

$$f_{k'}^{\mathrm{I},k} + \sum_{k''} f_{k',k''}^{\mathrm{B},k} - f_{k'}^{\mathrm{E},k} = 0, \quad k', k \in HCT,$$

$$f_{k'}^{\mathrm{O},k} + \sum_{k''} f_{k'',k'}^{\mathrm{B},k} - f_{k'}^{\mathrm{E},k} = 0, \quad k', k \in HCT,$$

$$T_k f_{k'}^{\mathrm{I},k} + \sum_{k''} f_{k',k''}^{\mathrm{B},k} t_{k''}^{\mathrm{O},k} - f_{k'}^{\mathrm{E},k} t_{k'}^{\mathrm{I},k} = 0, \quad k', k \in HCT,$$

$$q_{ij} - f_j^{\mathrm{E},i}(t_j^{\mathrm{I},i} - t_j^{\mathrm{O},i}), \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$q_{ij} - f_i^{\mathrm{E},j}(t_i^{\mathrm{O},j} - t_i^{\mathrm{I},j}), \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$dt_{ij}^1 = t_j^{\mathrm{I},i} - t_i^{\mathrm{O},j}, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$dt_{ij}^2 = t_j^{\mathrm{O},i} - t_i^{\mathrm{I},j}, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$dt_{ij}^1 \geqslant \mathrm{EMAT}, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$dt_{ij}^2 \geqslant \mathrm{EMAT}, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$LMTD_{ij} = \frac{dt_{ij}^1 - dt_{ij}^2}{\ln(dt_{ij}^1/dt_{ij}^2)}, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_j^{\mathrm{I},i} \geqslant 0, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_i^{\mathrm{I},j} \geqslant 0, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_{j,k}^{\mathrm{B},i} \geqslant 0, \quad i\in\mathcal{I}, j,k\in\mathcal{J},$$

$$f_{i,k}^{\mathrm{B},j} \geqslant 0, \quad i,k\in\mathcal{I}, j\in\mathcal{J},$$

$$f_j^{\mathrm{O},i} \geqslant 0, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_i^{\mathrm{O},j} \geqslant 0, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_j^{\mathrm{E},i} - F_i y_{ij} \leqslant 0, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_i^{\mathrm{E},j} - F_j y_{ij} \leqslant 0, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_j^{\mathrm{E},i} \geqslant \frac{q_{ij}}{\Delta T_{ij}^{\max}}, \quad i\in\mathcal{I}, j\in\mathcal{J},$$

$$f_i^{\mathrm{E},j} \geqslant \frac{q_{ij}}{\Delta T_{ij}^{\max}}, \quad i\in\mathcal{I}, j\in\mathcal{J}.$$

Theorem 4. *The HENS problem without decomposition (UMN) is $\mathcal{NP}$-hard in the strong sense.*

Proof 3. When problem *UMN* is restricted to the special case where $N_{\max} = |\mathcal{I}| + |\mathcal{J}| + 1$, $\beta_{ij} = 0$ for all combinations of i and j, $c_m = 0$ and $c_n = 0$ for all utilities, and with only one temperature interval ($K = 1$), as in the case of problem *MN1* in the proof of Theorem 3, the objective function becomes the following:

$$\min \sum_{i\in\mathscr{I}}\sum_{j\in\mathscr{J}} c_{ij}y_{ij}.$$

This case is only concerned with fixed costs of heat exchanger units. Also, since problem U is in $\mathscr{P}$, the minimum utilities cost problem may be solved in polynomial time using some of the parameters needed in UMN. Set lower and upper bounds L_{ij} and U_{ij} for either $i\in HU$ or $j\in$CU to the variable solutions Q_i^{HU} and Q_j^{CU} of problem U for the corresponding utility streams. This problem will be denoted as $UMN1$. Since the constraint set of U is a subset of constraints (15) of problem UMN, these upper and lower bounds produce a problem that remains feasible. Also, if these bounds are greater than zero, it is known that the integer variables for those utilities must be equal to 1 in the solution of $UMN1$, thus the integer variables may be removed from those bounding constraints as an equivalent formulation. Then, the remaining formulation is problem $MN1$. Therefore, $MN1$ is a special case of UMN. □

4.3. Stagewise simultaneous synthesis

Given:

- sets H of hot process streams and C of cold process streams;
- inlet and outlet temperatures and heat capacity flow rates for all process streams and utility temperatures;
- cost parameters for utilities, fixed and variable heat exchanger costs;
- overall heat transfer coefficients for heat exchangers;

the stagewise simultaneous synthesis formulation for HENS of Yee and Grossmann (1990) solves for the lowest total annual cost heat exchanger network. The simplifying assumptions of this HENS formulation are that mixing is isothermal, there are no split streams flowing through more than one exchanger, and there are no by-pass streams. In this formulation, there is no temperature interval partitioning as in problems MN and UMN, and the stages are used to increase the number of possibilities for network configurations:

Problem S

$$\min C^{\mathrm{CU}}\sum_{i\in H} q_i^{\mathrm{CU}} + C^{\mathrm{HU}}\sum_{j\in C} q_j^{\mathrm{HU}} + \sum_{i\in H}\sum_{j\in C}\sum_{k\in\mathscr{ST}} C_{ij}^{F} z_{ijk} + \sum_{i\in H} C_i^{F,\mathrm{CU}} z_i^{\mathrm{CU}} + \sum_{j\in C} C_j^{F,\mathrm{HU}} z_j^{\mathrm{HU}} +$$

$$+ \sum_{i\in H}\sum_{j\in C}\sum_{k\in\mathscr{ST}} C_{ij}^{A}\left[\frac{Q_{ijk}}{\left(\mathscr{U}_{ij}\left[(dt_{ijk}dt_{ijk+1})\frac{(dt_{ijk}+dt_{ijk+1})}{2}\right]^{\frac{1}{3}}\right)}\right]^{\beta_{ij}}$$

$$+ \sum_{i\in H} C_i^{A,\mathrm{CU}}\left[\frac{q_i^{\mathrm{CU}}}{\left(\mathscr{U}_i^{\mathrm{CU}}\left[dt_i^{\mathrm{CU}}\frac{(T_i^{\mathrm{OUT,H}}-T^{\mathrm{IN,CU}})\{dt_i^{\mathrm{CU}}+(T_i^{\mathrm{OUT,H}}-T^{\mathrm{IN,CU}})\}}{2}\right]^{\frac{1}{3}}\right)}\right]^{\beta_i^{\mathrm{CU}}}$$

$$+ \sum_{j\in C} C_j^{A,\mathrm{HU}}\left[\frac{q_j^{\mathrm{HU}}}{\left(\mathscr{U}_j^{\mathrm{HU}}\left[dt_j^{\mathrm{HU}}\frac{(T^{\mathrm{IN,HU}}-T_j^{\mathrm{OUT,C}})\{dt_j^{\mathrm{HU}}+(T^{\mathrm{IN,HU}}-T_j^{\mathrm{OUT,C}})\}}{2}\right]^{\frac{1}{3}}\right)}\right]^{\beta_j^{\mathrm{HU}}}$$

subject to

$$(T_i^{\mathrm{IN,H}}-T_i^{\mathrm{OUT,H}})F_i^{\mathrm{H}} = \sum_{k\in\mathscr{ST}}\sum_{j\in C} Q_{ijk} + q_i^{\mathrm{CU}}, \quad i \in H,$$

$$(T_j^{\mathrm{OUT,C}}-T_j^{\mathrm{IN,C}})F_j^{\mathrm{C}} = \sum_{k\in\mathscr{ST}}\sum_{i\in H} Q_{ijk} + q_j^{\mathrm{HU}}, \quad j \in C,$$

$$(t_{ik}^{\mathrm{H}}-t_{ik+1}^{\mathrm{H}})F_i^{\mathrm{H}} = \sum_{j\in C} Q_{ijk}, \quad i\in H,\ k \in \mathscr{ST},$$

$$(t_{jk}^{\mathrm{C}}-t_{jk+1}^{\mathrm{C}})F_j^{\mathrm{C}} = \sum_{i\in H} Q_{ijk}, \quad j \in C,\ k \in \mathscr{ST},$$

$$T_i^{\mathrm{IN,H}} = t_{i1}^{\mathrm{H}}, \quad i \in H,$$

$$T_j^{\mathrm{IN,C}} = t_{jK+1}^{\mathrm{C}}, \quad j \in C,$$

$$t_{ik}^{\mathrm{H}} \geqslant t_{ik+1}^{\mathrm{H}}, \quad i\in H,\ k \in \mathscr{ST},$$

$$t_{jk}^{\mathrm{C}} \geqslant t_{jk+1}^{\mathrm{C}}, \quad j \in C,\ k\in \mathscr{ST},$$

$$T_i^{\mathrm{OUT,H}} \leqslant t_{ik+1}^{\mathrm{H}}, \quad i \in H,$$

$$T_j^{\mathrm{OUT,C}} \geqslant t_{j1}^{\mathrm{C}}, \quad j \in C,$$

$$(t_{iK+1}^{\mathrm{H}}-T_i^{\mathrm{OUT,H}})F_i^{\mathrm{H}} = q_i^{\mathrm{CU}}, \quad i \in H,$$

$$(T_j^{\mathrm{OUT,C}}-t_{j1}^{\mathrm{C}})F_j^{\mathrm{C}} = q_j^{\mathrm{HU}}, \quad j \in C,$$

$$Q_{ijk}-\Omega z_{ijk} \leqslant 0, \quad i \in H,\ j \in C,\ k \in \mathscr{ST},$$

$$q_i^{\mathrm{CU}}-\Omega z_i^{\mathrm{CU}} \leqslant 0, \quad i \in H,$$

$$q_j^{\mathrm{HU}}-\Omega z_j^{\mathrm{HU}} \leqslant 0, \quad j \in C,$$

$$Q_{ijk} \geqslant 0, \quad i \in H,\ j \in C,\ k \in \mathscr{ST},$$

$$q_i^{\mathrm{CU}} \geqslant 0, \quad i \in H,$$

$$q_j^{\mathrm{HU}} \geqslant 0, \quad j \in C,$$

$$z_{ijk}\in\{0,1\} \quad i \in H,\ j\in C,\ k \in \mathscr{ST},$$

$$z_i^{\mathrm{CU}}\in\{0,1\} \quad i \in H,$$

$$z_j^{\mathrm{HU}}\in\{0,1\} \quad i \in C,$$

$$\left.\begin{array}{ll}
dt_{ijk} \leqslant t_{ik}^{\mathrm{H}}-t_{jk}^{\mathrm{C}}+\Gamma(1-z_{ijk}) & i\in H,\ j\in C,\ k\in\mathscr{ST} \\
dt_{ijk+1} \leqslant t_{ik+1}^{\mathrm{H}}-t_{jk+1}^{\mathrm{C}}+\Gamma(1-z_{ijk}) & i\in H,\ j\in C,\ k\in\mathscr{ST} \\
dt_i^{\mathrm{CU}} \leqslant t_{iK+1}^{\mathrm{H}}-T^{\mathrm{OUT,CU}}+\Gamma(1-z_i^{\mathrm{CU}}) & i\in H \\
dt_j^{\mathrm{HU}} \leqslant T^{\mathrm{OUT,HU}}-t_{j1}^{\mathrm{C}}+\Gamma(1-z_j^{\mathrm{HU}}) & j\in C \\
dt_{ijk} \geqslant \mathrm{EMAT} & i\in H,\ j\in C,\ k\in\mathscr{ST} \\
dt_i^{\mathrm{CU}} \geqslant \mathrm{EMAT} & i\in H \\
dt_j^{\mathrm{HU}} \geqslant \mathrm{EMAT} & j\in C
\end{array}\right\} \quad (16)$$

Let problem S be restricted to the special case with only one stage (i.e., $|\mathcal{ST}|=1$) and with the following parameter specifications for no utilities or exchanger area costs, and a restriction on the inlet and outlet temperatures of the process streams:

$$C^{\mathrm{CU}}=0$$

$$C^{\mathrm{HU}}=0$$

$$C^{\mathrm{A}}_{ij}=0, \quad i \in H, j \in C,$$

$$C^{\mathrm{A,CU}}_{i}=0, \quad i \in H,$$

$$C^{A,\mathrm{HU}}_{j}=0, \quad j \in C,$$

$$T^{\mathrm{OUT,H}}_{i} \geqslant T^{\mathrm{OUT,C}}_{j}+\mathrm{EMAT}, \quad i \in H, j \in C. \tag{17}$$

The resulting objective function is the following:

$$\min \sum_{i \in H} \sum_{j \in C} C^{\mathrm{F}}_{ij} z_{ij} + \sum_{i \in H} C^{\mathrm{F,CU}}_{i} z^{\mathrm{CU}}_{i} + \sum_{j \in C} C^{\mathrm{F,HU}}_{j} z^{\mathrm{HU}}_{j}. \tag{18}$$

With this new objective function, problem S constraints (16) become completely redundant since variables dt_{ijk}, dt^{CU}_{i}, and dt^{HU}_{j} no longer appear in the objective function, and due to restriction (17) these variables can be arbitrarily assigned to any value between the upper and lower bounds specified in constraints (16) for any value of the z-variables (for example, they can be set equal to EMAT). Therefore, constraints (16) may be removed. The remaining equivalent formulation will be denoted as problem $S1$. Problem $S1$ has objective function (18) and the following constraint equations:

$$(T^{\mathrm{IN,H}}_{i} - T^{\mathrm{OUT,H}}_{i})F^{\mathrm{H}}_{i} = \sum_{j \in C} q_{ij} + q^{\mathrm{CU}}_{i}, \quad i \in H,$$

$$(T^{\mathrm{OUT,C}}_{j} - T^{\mathrm{IN,C}}_{j})F^{\mathrm{C}}_{j} = \sum_{i \in H} q_{ij} + q^{\mathrm{HU}}_{j}, \quad j \in C,$$

$$\left.\begin{array}{ll}
(t^{\mathrm{H}}_{i1} - t^{\mathrm{H}}_{i2})F^{\mathrm{H}}_{i} = \sum_{j \in C} q_{ij} & i \in H \\
(t^{\mathrm{C}}_{j1} - t^{\mathrm{C}}_{j2})F^{\mathrm{C}}_{j} = \sum_{i \in H} q_{ij} & j \in C \\
T^{\mathrm{IN,H}}_{i} = t^{\mathrm{H}}_{i1} & i \in H \\
T^{\mathrm{IN,C}}_{j} = t^{\mathrm{C}}_{j2} & j \in C \\
t^{\mathrm{H}}_{i1} \geqslant t^{\mathrm{H}}_{i2} & i \in H \\
t^{\mathrm{C}}_{j1} \geqslant t^{\mathrm{C}}_{j2} & j \in C \\
T^{\mathrm{OUT,H}}_{i} \leqslant t^{\mathrm{H}}_{i2} & i \in H \\
T^{\mathrm{OUT,C}}_{j} \geqslant t^{\mathrm{C}}_{j1} & j \in C \\
(t^{\mathrm{H}}_{i2} - T^{\mathrm{OUT,H}}_{i})F^{\mathrm{H}}_{i} = q^{\mathrm{CU}}_{i} & i \in H \\
(T^{\mathrm{OUT,C}}_{j} - t^{\mathrm{C}}_{j1})F^{\mathrm{C}}_{j} = q^{\mathrm{HU}}_{j} & j \in C
\end{array}\right\} \tag{19}$$

$$q_{ij} - \Omega z_{ij} \leqslant 0, \quad i \in H, j \in C,$$

$$q^{\mathrm{CU}}_{i} - \Omega z^{\mathrm{CU}}_{i} \leqslant 0, \quad i \in H,$$

$$q^{\mathrm{HU}}_{j} - \Omega z^{\mathrm{HU}}_{j} \leqslant 0, \quad j \in C,$$

$$q_{ij} \geqslant 0, \quad i \in H, j \in C,$$

$$q^{\mathrm{CU}}_{i} \geqslant 0, \quad i \in H,$$

$$q^{\mathrm{HU}}_{j} \geqslant 0, \quad j \in C,$$

$$z_{ij} \in \{0,1\}, \quad i \in H, j \in C,$$

$$z^{\mathrm{CU}}_{i} \in \{0,1\}, \quad i \in H,$$

$$z^{\mathrm{HU}}_{j} \in \{0,1\}, \quad j \in C.$$

Since none of the temperature variables or utility heat load variables appear in the objective function of problem $S1$, constraints (19) also become redundant. They are satisfiable by any set of heat load variables that could be produced by the remaining constraints. The temperature variables are totally dependent on the heat load variables. Removal of these constraints produces an equivalent formulation with the exact same objective function and constraint set as problem $M1$. Therefore, $M1$ is a special case of S.

Theorem 5. *The stagewise simultaneous HENS problem is $\mathcal{NP}$-hard in the strong sense.*

5. Conclusions

Although HENS has been one of the most studied problems in process synthesis, even small HENS problems have not been solved to global optimality to date. In fact, even finding feasible solutions using simultaneous synthesis methods has been troublesome. The largest problem in stagewise simultaneous HENS in which a feasible solution was found has only seven process streams (Yee and Grossmann, 1990). A local optimum for a ten stream problem was found only by placing additional restrictions on the problem formulation (Daichendt and Grossmann, 1994a,b). The difficulty of solving large HENS problems can now be explained under the light of the theoretical results of our paper.

Through a computational complexity analysis, we have proven that even simple special cases of HENS are $\mathcal{NP}$-hard in the strong sense. This implies that the existence of a computationally efficient exact solution algorithm is highly unlikely to exist for this problem. Our proofs were based on showing that the 3-PARTITION problem is embedded in various forms of the HENS problem as shown in Fig. 4. A different approach, via restriction of the HENS problem to the $\mathcal{NP}$-hard Knapsack problem, is presented in the Appendix A. The proofs suggest that heuristic algorithms that have performed well for the 3-PARTITION and the Knapsack may be found useful when extended to

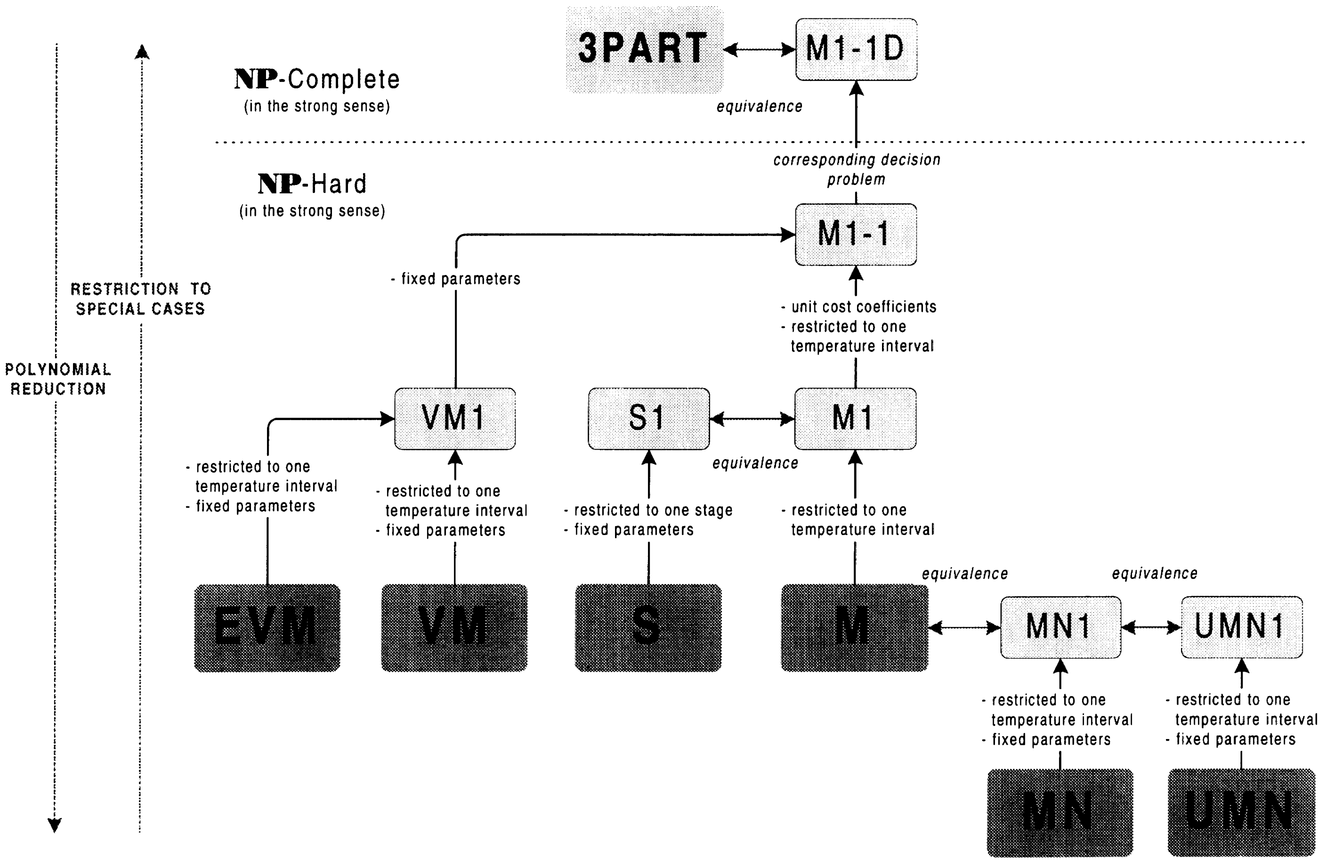

Fig. 4. Polynomial reducibility of the 3-partition problem to various HENS problems.

Table 1
Complexity results

Problem	Target	Source	Complexity
UU	Utilities	Papoulias and Grossmann (1983)	$\mathscr{P}$
U	Utilities	Papoulias and Grossmann (1983)	$\mathscr{P}$
U'	Utilities	Cerda, Westerberg, Mason, and Linnhoff (1983)	$\mathscr{P}$
M	Matches	Papoulias and Grossmann (1983)	$\mathscr{NP}$-hard in the strong sense
M'	Matches	Cerda and Westerberg (1983)	$\mathscr{NP}$-hard in the strong sense
VM	Matches	Gundersen and Grossmann (1990)	$\mathscr{NP}$-hard in the strong sense
EVM	Matches	Gundersen, Duvold, and Hashemi-Ahmady (1996)	$\mathscr{NP}$-hard in the strong sense
MN	Match-network	Floudas and Ciric (1989)	$\mathscr{NP}$-hard in the strong sense
UMN	Simultaneous	Ciric and Floudas (1991)	$\mathscr{NP}$-hard in the strong sense
S	Simultaneous	Yee and Grossmann (1990)	$\mathscr{NP}$-hard in the strong sense

HENS. Our computational complexity results are summarized in Table 1 and are evidence of the computational complexity of the general process synthesis problem.

The proof that this problem is hard should not be construed as a deterrent to future research in this field. These complexity results should provide motivation for the further development of specialized optimization algorithms, heuristics, and approximations for HENS. In fact, results such as these provide the required license to develop approximation schemes which run in polynomial time and have performance guarantees for the quality of the solutions they produce. Examples of such approximation schemes and a detailed discussion of surrounding issues are provided in Liu and Sahinidis (1997) and Ahmed and Sahinidis (2000).

Acknowledgements

The authors wish to thank Dr. Udatta Palekar and Dr. Shabbir Ahmed for helpful comments and suggestions. The authors are also grateful to Dr. William R. Johns whose comments motivated us to improve the presentation so as to make the paper more accessible to the audience. Partial financial support from the National Science Foundation under grant CTS-9704643 is gratefully acknowledged.

Appendix A. $\mathscr{NP}$-hardness proof via restriction to the Knapsack problem

In this appendix, we prove that the HENS problem is $\mathscr{NP}$-hard via restriction to the Knapsack problem. Since Knapsack has a known pseudo-polynomial time solution algorithm, the proofs of Sections 3 and 4are stronger than the ones here, as they showed that HENS is $\mathscr{NP}$-hard *in the strong sense*. Nonetheless, the appendix is provided to illustrate some of the properties of HENS that liken it to the Knapsack problem. Such properties can be exploited in the design of exact and approximate algorithms for HENS.

Let problem M be restricted to problem $M1$ as in Section 3.2. Problem $M1$ can be restricted by setting $c_{ij}=0$ for $i\in\mathscr{I}, j\in\mathscr{J}\setminus\{j'\}$ where j' is one specific element of $\mathscr{J}$. This case is an instance of the minimum number of matches problem in which all the inlet temperatures of hot streams are equal to the outlet temperatures of the cold streams and vice versa, causing the existence of only one temperature interval. Also, heat exchangers for all but one type of match are in possession or are very inexpensive and, thus, only one type of heat exchanger would be undesirable and have a nonzero cost coefficient. See Fig. 2 for a visual depiction of this case.Then, the problem becomes:

Problem M2

$$\min \sum_{i\in\mathscr{I}} c_{ij'} y_{ij'}$$

subject to

$$\sum_{j\in\mathscr{J}} q_{ij} = s_i, \quad i\in\mathscr{I},$$

$$\sum_{i\in\mathscr{I}} q_{ij} = d_j, \quad j\in\mathscr{J}, \tag{A.1}$$

$$q_{ij} \leqslant U_{ij} y_{ij}, \quad i\in\mathscr{I},\ j\in\mathscr{J}, \tag{A.2}$$

$$q_{ij} \geqslant 0, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

$$y_{ij}\in\{0,1\}, \quad i\in\mathscr{I},\ j\in\mathscr{J},$$

Given a finite set $\mathscr{I}=\{1,\ldots,m\}$, a size $S_i\in\mathscr{Z}^+$, a value $C_i\in\mathscr{Z}^+$, and a bound $b\in\mathscr{Z}^+$ the minimum binary Knapsack problem (Papadimitriou and Steiglitz, 1982) is as follows:

Problem KP

$$\min \sum_{i\in\mathscr{I}} C_i Y_i$$

subject to

$$\sum_{i\in\mathscr{I}} S_i Y_i \geqslant b, \tag{A.3}$$

$$Y_i\in\{0,1\}, \quad i\in\mathscr{I}.$$

Theorem 6. *Problem M2 has a feasible solution if and only if problem KP has a feasible solution.*

Proof 4. *Necessity*. Consider an instance of $M2$ and construct an instance of KP by setting $C_i=c_{ij'}$, $b=d_{j'}$ and $S_i=s_i$. Let $\{q_{ij}|i\in\mathscr{I}, j\in\mathscr{J}\}$ and $\{y_{ij}|i\in\mathscr{I}, j\in\mathscr{J}\}$ be a feasible solution to $M2$ and consider $Y_i=y_{ij'}$ as a candidate solution for problem KP. By the definition of U_{ij}, it is known that $s_i\geqslant U_{ij}$ for all $i\in\mathscr{I}$. From (20), it is known that $\Sigma_{i\in\mathscr{I}} q_{ij}=d_{j'}=b$. Then

$$\sum_{i\in\mathscr{I}} S_i Y_i = \sum_{i\in\mathscr{I}} s_i y_{ij'} \geqslant \sum_{i\in\mathscr{I}} U_{ij'} y_{ij'} \geqslant \sum_{i\in\mathscr{I}} q_{ij'} = d_{j'} = b$$

from (21). Therefore, $\{Y_i|Y_i=y_{ij'}\}$ is a feasible solution to *KP.Sufficiency*. Consider an instance of KP and construct an instance of $M2$ by setting $s_i=S_i$, $c_{ij'}=C_i$, $d_{j'}=b$ and note that (1) is predetermined to be true. Let $\{Y_i|i\in\mathscr{I}\}$ be a feasible solution to this instance of the minimum binary Knapsack problem. Consider $y_{ij'}=Y_i$ as a portion of a candidate solution for problem $M2$. Let $\mathscr{I}_0=\{i\in\mathscr{I}|y_{ij'}=0\}$ and $\mathscr{I}_1=\{i\in\mathscr{I}|y_{ij'}=1\}$. Then, let $q_{ij'}=0$ for all $i\in\mathscr{I}_0$. The remaining feasibility constraints of $M2$ can then be reformulated into an LP. To guarantee (1), some of the sources in the reformulation need to be redefined since $\Sigma_{i\in\mathscr{I}} s_i y_{ij'}$ may be greater than $d_{j'}$ due to (22) in *KP* and thus $q_{ij'}$ cannot be arbitrarily set equal to $\{s_i|y_{ij'}=1\}$. Let

$$s_i' = \begin{cases} \max\left\{s_i - \dfrac{d_{j'}}{H}, 0\right\} & \text{for } i\in\mathscr{I}_1, \text{ where } H=\displaystyle\sum_{i\in\mathscr{I}_1} i, \\ s_i & \text{otherwise.} \end{cases}$$

Then, let $q_{ij'}=s_i-s_i'$ for all $i\in\mathscr{I}_1$. The feasibility equation (1) becomes

$$\sum_{i\in\mathscr{I}} s_i' = \sum_{j\in\mathscr{J}, j\neq j'} d_j, \tag{A.4}$$

and the formulation of the remaining constraints is as follows:

Find a set $\{q_{ij}|i\in\mathscr{I}, j\in\mathscr{J}, j\neq j'\}$ such that

$$\sum_{j\in\mathscr{J}} q_{ij} = s_i, \quad i\in\mathscr{I},$$

$$\sum_{i\in\mathscr{I}} q_{ij} = d_j, \quad j\in\mathscr{J}, j\neq j',$$

$$U_{ij} \geqslant q_{ij} \geqslant 0, \quad i\in\mathscr{I}, j\in\mathscr{J}, j\neq j'.$$

This LP formulation is a transportation problem. Considering that (1) is true for any instance of $M2$, a feasible solution to this transportation problem is guaranteed by the Northwest Corner rule (Wagner, 1975) due to (23). Then, set:

$$y_{ij} = \begin{cases} Y_i & \text{for } j = j', \\ 0 & \text{for } j \neq j',\ q_{ij} = 0, \\ 1 & \text{otherwise,} \end{cases}$$

and a feasible solution to $M2$ is produced. □

Corollary 5. *The optimal solution of M2 is equal to the optimal solution to KP.*

It is well known that KP is $\mathcal{NP}$-hard (Papadimitriou and Steiglitz, 1982). Then, $M2$ is $\mathcal{NP}$-hard because $M2$ is equivalent to KP. This implies that M is also $\mathcal{NP}$-hard.

Corollary 6. *The minimum number of matches problem is* $\mathcal{NP}$*-hard.*

Then, the proofs of Corollary 4 and Theorem 3, Theorem 4 and Theorem 5 imply that:

Corollary 7. *The following problems are* $\mathcal{NP}$*-hard:*
1. *The vertical heat transfer minimum number of matches problem (VM) and its extended version (EVM).*
2. *The simultaneous match-network problem (MN).*
3. *The HENS problem without decomposition (UMN).*
4. *The stagewise simultaneous synthesis HENS problem (S).*

References

Ahmed, S., & Sahinidis, N. V. (2000). Analytical investigations of the process planning problem. *Computers and Chemical Engineering*, *23*(11–12), 1605–1621.

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Upper Saddle River, NJ: Prentice Hall.

Biegler, L. T., Grossmann, I. E., & Westerberg, A. W. (1997). *Systematic methods of chemical process design*. Upper Saddle River, NJ: Prentice Hall.

Braatz, R. D., & Russell, E. L. (1999). Robustness margin computation for large scale systems. *Computers and Chemical Engineering*, *23*(8), 1021–1030.

Cerda, J., & Westerberg, A. W. (1983). Synthesizing heat exchanger networks having restricted stream/stream matches using transportation problem formulations. *Chemical Engineering Science*, *38*(10), 1723–1740.

Cerda, J., Westerberg, A. W., Mason, D., & Linnhoff, B. (1983). Minimum utility usage in heat exchanger network synthesis. *Chemical Engineering Science*, *38*(3), 373–387.

Ciric, A. R., & Floudas, C. A. (1991). Heat exchanger network synthesis without decomposition. *Computers and Chemical Engineering*, *15*(6), 385–396.

Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of the third ACM symposium on theory of computing*, pp. 151–158.

Daichendt, M. M., & Grossmann, I. E. (1994a). A preliminary screening procedure for MINLP heat exchanger network synthesis using aggregated models. *Chemical Engineering Research and Design*, *72*(A), 357–363.

Daichendt, M. M., & Grossmann, I. E. (1994b). Preliminary screening procedure for the MINLP synthesis of process systems-II. Heat exchanger networks. *Computers and Chemical Engineering*, *18*(8), 679–709.

Floudas, C. A., & Ciric, A. R. (1989). Strategies for overcoming uncertainties in heat exchanger network synthesis. *Computers and Chemical Engineering*, *13*(10), 1133–1152.

Floudas, C. A., Ciric, A. R., & Grossmann, I. E. (1986). Automatic synthesis of optimum heat exchanger network configurations. *American Institute of Chemical Engineers Journal*, *32*(2), 276–290.

Furman, K. C., and Sahinidis, N. V. An annotated bibliography for heat exchanger network synthesis in the 20th century, submitted for publication.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. New York: W.H. Freeman.

Gundersen, T., & Grossmann, I. E. (1990). Improved optimization strategies for automated heat exchanger network synthesis through physical insights. *Computers and Chemical Engineering*, *14*(9), 925–944.

Gundersen, T., & Naess, L. (1988). The synthesis of cost optimal heat exchanger networks. *Computers and Chemical Engineering*, *12*(6), 503–530.

Gundersen, T., Duvold, S., & Hashemi-Ahmady, A. (1996). An extended vertical MILP model for heat exchanger network synthesis. *Computers and Chemical Engineering*, *20*(Suppl.), S97–S102.

Ibarra, O. H., & Kim, C. E. (1975). Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the Association for Computing Machinery*, *22*, 463–468.

Ježowski, J. (1994a). Heat exchanger network grassroot and retrofit design. The review of the state-of-the-art: Part I. Heat exchanger network targeting and insight based methods of synthesis. *Hungarian Journal of Industrial Chemistry*, *22*, 279–294.

Ježowski, J. (1994b). Heat exchanger network grassroot and retrofit design. The review of the state-of-the-art: Part II. Heat exchanger network synthesis by mathematical methods and approaches for retrofit design. *Hungarian Journal of Industrial Chemistry*, *22*, 295–308.

Karmarkar, N. (1984). A new polynomial time-algorithm for linear programming. *Combinatorica*, *4*, 373–395.

Khachian, L. G. (1979). A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, *20*, 191–194.

Klee, V., & Minty, G. J. (1972). How good is the simplex algorithm? In O. Shisha, *Inequalities III*. New York: Academic Press.

Lewis, H. R., & Papadimitriou, O. H. (1998). *Elements of the theory of computation*. Upper Saddle River, NJ: Prentice Hall.

Linnhoff, B., & Flower, J. R. (1978). Synthesis of heat exchanger networks I. Systematic generation of energy optimal networks. *American Institute of Chemical Engineers Journal*, *24*(4), 633–642.

Liu, M. L., & Sahinidis, N. V. (1997). Bridging the gap between heuristics and optimization: capacity expansion case. *American Institute of Chemical Engineers Journal*, *43*(9), 2289–2299.

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and combinatorial optimization*. New York: Wiley.

Papadimitriou, C. H., & Steiglitz, X. (1982). *Combinatorial optimization: algorithms and complexity*. Englewood Cliffs, NJ: Prentice-Hall.

Papoulias, S. A., & Grossmann, I. E. (1983). A structural optimization approach in process synthesis – II. Heat recovery networks. *Computers and Chemical Engineering*, *7*(6), 707–721.

Saboo, A. K., Morari, M., & Colberg, R. D. (1986). RESHEX: an interactive software package for the synthesis and analysis of resilient heat-exchanger networks – II. *Computers and Chemical Engineering*, *10*(6), 591–599.

Wagner, H. M. (1975). *Principles of operations research* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Yee, T. F., & Grossmann, I. E. (1990). Simultaneous optimization models for heat integration – II. Heat exchanger network synthesis. *Computers and Chemical Engineering*, *14*(10), 1165–1184.