KNOWLEDGE-BASED APPROACH FOR PROCESS SYNTHESIS AUTOMATION:
AN APPLICATION TO HEAT EXCHANGER NETWORK SYNTHESIS

by

CHETAN MEHTA

B. Tech., Indian Institute of Technology, Bombay, 1982.

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Chemical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1986

Approved by

L. T. Fan

Major Professor

ACKNOWLEDGEMENTS

I wish to express my gratitude and sincere thanks to my advisor, Dr. L. T. Fan, for his continuous support and encouragement throughout the course of this work. At times, when I reached dead-ends in carrying out this research, it was his encouragement and support that provided me the strength to continue the work. This thesis would have been far inferior to its present form, but for his high standards in manuscript preparation and meticulous attention to detail. Thanks are also due to the committee members, Dr. J. C. Mathews and Dr. W. P. Walawender.

I am indebted to my friend and colleague, Mr. Sujeet Shenoi, for introducing me to the fascinating fields of Artificial Intelligence and Knowledge Engineering. His critiques and suggestions, in addition to the numerous discussion sessions we have shared, have contributed immensely to this work. I also wish to express my gratitude to Dr. Roger T. Hartley, Department of Computer Science, for providing me with the outlook and insights to the fields of Artificial Intelligence and Knowledge Engineering that are reflected in this work.

The financial support provided by the Engineering Experiment Station, Kansas State University and the National Science Foundation (grant DMC-8516870) is greatly acknowledged.

I wish to acknowledge the encouragement and support that my fiancee, Falguni Bhuta, has provided me during the preparation of this thesis. Finally, I dedicate this work to my parents, Mrs. and Mr. D. H. Mehta, without whose support and encouragement throughout my education, I would not have been in a position to write this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Design is a creative process by which we conceptualize and specify products, processes, and their interconnections to perform desired functions. All phases and disciplines of engineering involve design; yet it has been relatively neglected and poorly understood area. Almost all current knowledge of engineering design is embeded in human expert designers, who are usually quite unaware of the precise nature and form of knowledge they possess. Thus, considerable work is needed to extract, organize, and apply this knowledge.

Attempts at computerizing the design process have been primarily in the detailed design phase and not in the critical concept generation phase. Clearly, the quality of the final product is limited by the quality of the basic concept. The research efforts, therefore, should be directed towards evolving the basic principles and axiomatic systems for design and formalization of design methodologies, leading to the conceptual and theoretical frameworks for design. Attempts should be made to characterize, model, and quantify the designers' thought processes involved in representation of problems, organization of information, and visualization, conception as well as evaluation of new designs. Understanding the role of engineering heuristics in making the design decisions is crucial to have effective interaction of designers with the automated design and analysis tools. To accomplish these seemingly formidable tasks, a ***knowledge-based approach*** is inevitable.

A knowledge-based approach to problem solving involves extraction and formalization of the underlying knowledge behind any cause-effect relationship, and devising suitable schemes to represent and manipulate this knowledge. Thus, the knowledge-based approach relies upon the domain-specific knowledge, rather than a generalized solution methodology, to solve complex problems. This approach to problem solving has resulted from the successes over the past decade in *Artificial Intelligence (AI)* research; it has given rise to a new discipline, *Knowledge Engineering* (see, e.g., Barr and Feigenbaum, 1981). The knowledge-based approach enables us to build a computational model to solve a problem. This computational model closely mimics the reasoning process of a domain expert and hence, acts as an unbiased vehical for experimenting with various theories and strategies for the design process. Additionally, it leads to automated design systems that can perform on par with the domain experts. As demonstrated by its successes in several application areas, AI and Knowledge Engineering are capable of dealing with the type of problems encountered in design automation. For example, the knowledge-based approach has been successfully applied to integrated circuit (IC) design, including the LSI and VLSI design. It is the purpose of this work to demonstrate that such an approach can lead to significant progress in automating the design of chemical processes.

Design of a chemical process is an iterative procedure, typically consisting of four stages: synthesis, analysis, evaluation and optimization. Synthesis is the conceptualization step involving the specification of physical and chemical operations as well as the selection and interconnection of units (equipment) for implementing these operations to produce the desired physico-chemical transformations. Since the advent of the computer era, significant progress has been made in the development of more scientific and less empirical framework for modeling, simulating and optimizing the parameters for a given design. However, the generation of a processing structure or a flowsheet, a necessary prerequisite of design, remains largely a creative art (Siirola, 1982). The importance of correct structural choices in meeting process objectives has been demonstrated by the resultant economic benefits . This realization has led to the present efforts to investigate the possibility of formalizing the synthesis activity. It is worth noting that the ultimate goal of process synthesis automation is not merely to invent technically feasible designs; it is also to produce structural configurations that when analyzed, evaluated and optimized, will prove to be superior to (possibly all) other structural arrangements on the basis of the design objective criteria (Siirola, 1982).

Some efforts at automating the synthesis step have resulted in systems which have performed at levels far below that of even a novice design engineer (e.g., AIDES and BALTAZAR systems). This failure of machines to synthesis a process at an expert level can be ascribed to the fact that the synthesis step demands extensive knowledge from diverse areas within Chemical Engineering as well as from various other disciplines. Furthermore, the nature of the synthesis step is such that it requires the conceptual and symbolic manipulation of domain-specific knowledge in contrast to the numerical manipulation of "data" required for the other steps of process design. Consequently, if any significant progress is to be made in automating this step, a knowledge-based approach is inevitable. This work attempts to demonstrate how a knowledge-based approach can be employed to automate the task of process synthesis. The approach is applied to the problem of heat exchanger network synthesis.

ORGANIZATION OF THESIS

Chapter 2 provides an introduction to process synthesis automation and Knowledge Engineering. The chapter begins with a discussion of the nature and scope of process synthesis, followed by the analysis of the impact of computers on process synthesis. Next, the definitions and scopes of Artificial Intelligence (AI) and Knowledge Engineering are presented, along with the knowledge Engineering issues relevant to the automation of process synthesis. Finally, the classification of synthesis problems is provided, and a specific subproblem of synthesizing heat exchanger networks (HEN's) is chosen for demonstrating the use of the knowledge-based approach.

Chapter 3 contains a comprehensive review of the existing literature and the solution methodologies for the selected subproblem of heat exchanger network synthesis. The chapter is comprised of three parts; the first part discusses the problem specifications, the second part deals with the three steps of heat exchanger network (HEN) synthesis including preanalysis, network invention, and evolutionary modifications, and the third and the last part describes the problem representation scheme employed in the present work.

Chapters 4 and 5 are concerned with the extraction and formalization of knowledge for HEN synthesis. An elimination strategy to generate HEN's with the minimum number of units is presented in Chapter 4. The effect of a pinch point is analyzed and the necessary and sufficient conditions for elimination are derived. Chapter 5 discusses the need for stream splitting, with the help of an illustrative example. Stream splitting for this example problem is carried out as an exercise. Finally, the task of stream splitting is formalized, and a systematic procedure that employs the elimination conditions is proposed for carrying out stream splitting.

Schemes to represent and manipulate the HEN synthesis knowledge (formalized in Chapters 4 and 5) are proposed in Chapter 6. By resorting to these schemes, a systematic procedure is formulated for synthesizing a HEN featuring the minimum utility consumption and the minimum number of units. The procedure is demonstrated using an illustrative example. Finally, the conceptual design of a knowledge-based system for HEN synthesis is proposed; it utilizes the extracted knowledge as well as the proposed representation and manipulation schemes. A summary of accomplishments and the recommendations for future work are provided in the last chapter.

CHAPTER 2. KNOWLEDGE ENGINEERING AND PROCESS SYNTHESIS AUTOMATION

The activities related to process synthesis have a long history; these activities are involved in process development, design and modification, all of which play essential roles in the chemical and allied industries. Nevertheless, the development of systematic and formal approaches to process synthesis dates back only to 1968, except for the isolated treatment of selected subproblems, viz., Heat Exchanger Network Synthesis and Synthesis of Multicomponent Distillation Sequences (see, e.g., Hendry, et al., 1973). Thus process synthesis as a research area is barely two decades old. During these years, significant progress has been made, mostly in solving individual and specific problems. Substantial results are yet to be obtained in solving the all-encompassing, general problem of synthesizing complex processes in their entirety.

In this chapter, we shall briefly examine the scope of process synthesis and the impact of computers on this field. Recent developments in the field of computer science, specifically in Artificial Intelligence and Knowledge Engineering, have necessitated the development of an improved and innovative approach to computer-aided process synthesis. A brief discussion on this is followed by a review of some of the relevant issues of this new approach, a ***knowledge-based approach***; it offers a promising future for the automation of process synthesis. Finally, the application of this knowledge-based approach to process synthesis is discussed.

NATURE AND SCOPE OF PROCESS SYNTHESIS

Several authors have described the field of synthesis in diverse ways. Some of these descriptions are reproduced here.

The development of an industrial process requires skills in synthesis and analysis. In the words of Webster, synthesis is "the combination of often diverse conceptions into a coherent whole" and analysis is "an examination of a complex, its elements and their relations. In the words of Herbert A. Simon, "synthesis deals with how to make artifacts that have desired properties; analysis deals with how things are and how they work" (Rudd, et al., 1973).

According to Siirola and Rudd (1971), *process synthesis begins by the discovery, in the laboratory, of a sequence of chemical reactions which link readily available raw materials to more valuable products; it ends with the development of the flowsheet for the commercial process which exploits the chemistry most successfully.*

Westerburg (1980) states that the activity of synthesis occurs throughout a design, from the original process conception to the construction and operation, all involving discrete decision-making. He regards process synthesis as *a nonlinear, mixed integer and continuous variable optimization problem*, in which selection of the building blocks of the process and their interconnections is formulated as a set of discrete decisions represented by zero/one variables. For a particular set of discrete decisions, we must determine the optimal operating level for the corresponding process structure.

Nishida et al. (1981), in their review paper, have defined process synthesis as an act of determining the optimal type and design of the units within a process system; the interconnections of processing units generate the structure of the system. According to them, even when the desired performance of the system is specified, the structure of the system and the performance levels of the processing units are not determined uniquely; the task of synthesis is to select a particular structure out of a large number of alternatives which meet the specified performance requirements.

Several additional versions of the definition of process synthesis can be found in the literature (see, e.g., Umeda, 1983). Nevertheless, the descriptions given in the preceding paragraphs adequately represent the entire spectrum. These descriptions clearly fall into two distinct categories: the first two descriptions represent the early age of the field of process synthesis and the remaining two represent the more mature, current phase of process synthesis research. The two sets of descriptions, almost a decade apart, differ substantially in establishing the nature and scope of process synthesis. The later two descriptions suggest that the synthesis problem is one of selection of a particular structure out of several alternatives. The problem is formulated as an optimization problem, implying the presence of a "*superstructure*", i.e., a set of all possible building blocks of the process structure and all possible "*legal*" connections are specified at the "onset" of the synthesis. The task of synthesis is then to select an optimal subset of building blocks and their interconnections. In contrast, the first two descriptions view the synthesis problem to be one of conception of a feasible structure with specified constraints. A broader scope of synthesis is implied in that it involves decision-making at a higher level; instead of selecting a subset from a pre-specified, "*legal*" or feasible set of connections, decisions have to be made regarding what building blocks and connections are needed to accomplish the desired task (i.e., determining what are the "*legal*" building blocks and connections). The difference in viewpoints between the two sets of definitions has significant implications, which will be explored in subsequent sections. For the present, it will suffice to state that the "newer" or the more recent of the two opinions has a distinct flavor of computerization and is widely accepted among the researchers.

It should be noted, however, that until recently the use of computers was restricted to numeric computing only. Thus selection based on performance evaluation could be easily carried out by computers, but conceiving a structure, which requires symbolic, conceptual manipulation, could not be accomplished by the machines. This constraint seems to have substantially influenced the "newer" or more recent definitions and scopes for the field of process synthesis. Throughout this work, the earlier, broader descriptions will be followed and accepted, primarily because it is the aim of this work to demonstrate the feasibility of creating computer-aided process synthesis systems that fit into this broader scope of the synthesis description.

COMPUTERS IN PROCESS SYNTHESIS

The use of computers at various stages of process engineering work is widely accepted by engineers. In fact, computers have become such an integrated part of process synthesis that "process synthesis" has essentially become synonymous with "computer-aided process synthesis." Consequently, it is no surprise that most of the synthesis techniques developed over the years are primarily oriented towards computer usage.

In the 1960's, when the research on formalizing synthesis techniques was in its infancy, process design including the synthesis activities was carried out by engineers with the aid of computers; with the availability of time-sharing computer systems with on-line input/output devices, the interactive usage of computers had already become widely prevalent by that time. In such an interactive environment, the evolutionary steps of process synthesis were performed alternately by man and machine; the engineer's role was to give the information on process structure to the machine, which then analyzed the given structure and reported its performance evaluation back to the engineer for decision-making. Thus computers performed tbe extensive numeric calculations needed for analysis and simulation, whereas the task of "*actual synthesis*" was accomplished by the engineers.

The decade of the 1970's saw the proliferation of new, formal, systematic procedures and algorithms for solving the problems from various subgroups of synthesis, e.g., heat exchanger network synthesis and synthesis of separation sequences. Each of the subgroups has special characteristics, thus requiring different techniques for solution. These techniques can be classified into the following broad categories (see, e.g., Nishida et al., 1981; Umeda, 1983).

(a) Algorithmic methods including linear and non-linear programming, dynamic programming, mixed integer and continuous variable programming, etc.

(b) Search methods, such as total enumeration, depth first, breadth first and branch and bound.

(c) Evolutionary methods, e.g., parametric optimization.

(d) Heuristic methods based on a set of thumb-rules developed from past experience.

However, many of the latest synthesis methods do not fall strictly under any one category; they combine the principles of two or more categories, e.g., algorithmic evolutionary methods, heuristic evolutionary methods, etc.

Almost all the techniques, except the heuristic ones, involve extensive numeric computations; naturally, they would necessitate the use of computers. They take advantage of the fact that computers can crunch numbers several orders of magnitude faster than humans, thereby drastically reducing the necessary computational time. However, all these techniques require pre- and/or post-processing by engineers. Preprocessing involves transformation of a synthesis problem into an abstract computational model which can then be processed by a machine. Post-processing involves the interpretation of the results supplied by the machine. Except for extremely simple test cases, this processing tends to be rather involved and complex, so much so that at times it is "*easier*" to solve the problem "*manually*", without resorting to any abstract computational model. One of the most severe disadvantages of using an abstract model is that it behaves like a black-box; it is not possible to establish its "*goodness*" without exhaustive testing. We can not guarantee that the model will generate "good" solutions for each "new" problem it solves, even after testing it successfully with several problems. Furthermore, if the performance of a model is not up-to-the-mark, then it has to be "thrown away", to analyze its shortcomings and upgrade it is almost impossible. This is because the model is so far removed from the physical aspects of the problem that it is extremely difficult to relate the model to the existing and emerging knowledge as well as the insights and experience of an expert process engineer.

Techniques involving search strategies have severe limitations on the size of problems they can solve. Even for a moderate size problem, the combinatorial explosion generates a very large search space. In addition, evaluation of each possible solution (a node) in the search tree tends to be rather cumbersome. Moreover, generating the possible solutions for complex problems like synthesizing entire process flowsheet requires a great deal of knowledge about the problem domain and the experience and insight into problem solving strategies. These limitations have forced researchers to abandon "pure" search methods altogether. In case it is employed, a search strategy is almost always coupled with other approaches, model based (algorithmic) or heuristic.

Parametric methods suggest an optimization approach to synthesis. They require as a starting point, a super-structure which is a superset of all possible candidate solutions. To generate such a super-structure, we need to resort to some other synthesis techniques.

Consequently, these methods should more suitably be labeled optimization techniques and not synthesis techniques.

Heuristic methods are useful in that they usually give reasonable solutions, but the quality of solution is not guaranteed. Process engineers are more comfortable with such methods because these methods reflect, to a certain extent, the knowledge that the engineers themselves possess. The advantage of using such methods is that they can be easily updated when more knowledge is available to solve the problem. Also, if a method fails to give reasonable results, the methods is amenable to analysis, upgrading and modification. However, when computerized, these methods tend to lose their power because the conventional computerization transforms the method into an algorithmic one. Moreover, an engineer uses his judgement, common-sense knowledge and his knowledge about the problem domain in conjunction with the heuristics. In a computerized version, the heuristics work in isolation resulting sometimes in totally ridiculous solutions.

All avenues, explored so far by researchers in computerizing process synthesis, have led to blind alleys; even though sufficient knowledge exists for process engineers to conceive and synthesize process flowsheets, automated synthesis systems are still very far from reality. Consequently, the synthesis activities in process industries is still carried out using man-machine systems where the machines carry out the analysis and simulation and the men provide the decision-making ability so crucial for synthesis and design tasks. Essential to a competent designer is the ability to make decisions in the presence of incomplete, uncertain or fuzzy information. This necessitates the use of fundamental concepts and "standard" procedures (heuristics or thumb-rules). This pool of information, termed as ***domain-specific knowledge***, is largely missing in the current design and synthesis systems. If we are to have computer-aided process synthesis systems that utilize machines effectively so as to minimize the human work load, then the domain-specific knowledge and the ability to make decisions based on this knowledge must be imparted to the machines. This will give rise to computer-aided process synthesis systems whose performance level will be comparable to that of human experts, in other words, systems that will be "***trusted***" by the process engineers.

Though it may sound over ambitious, this task has already been accomplished in numerous fields. Several systems, employed routinely, perform tasks that are thought to require human intelligence and creativity. Some of the well-known systems are: MYCIN, CASNET, CADUCEUS, and PUFF for medical diagnosis, DENDRAL for chemical structure elucidation, XCON for computer configuration, PROSPECTOR for mineral exploration and DIPMETER ADVISOR for oil well log interpretation (see, e.g., Barr and Feigenbaum, 1981). These systems have demonstrated performance levels that rival or exceed those of human experts in their respective domains. Building such computer a system falls under the purview of the emerging field of *Knowledge Engineering* which was born out of the research in *Artificial Intelligence*. The success of the Knowledge Engineering approach to problem solving gives us a ray of hope and the promise of one day having a fully automated process synthesis system. To construct such a system, it is most appropriate to begin with an overview of Knowledge Engineering and Artificial Intelligence.

ARTIFICIAL INTELLIGENCE AND KNOWLEDGE ENGINEERING

Artificial Intelligence (AI) deals with making machines emulate intelligent human behavior. According to Barr and Feigenbaum (1981), AI is a branch of computer science which deals with designing computer systems that exhibit some of the characteristics usually associated with intelligent human behavior: understanding natural (common) languages, learning, reasoning, solving complex problems, and so on. Rich (1983) gives an interesting description of AI: "*It is the study of how to make machines do things at which, at the moment, people are better.*"

In the 1970's, researchers in AI recognized that human intelligence is domain specific, and that intelligence has the form of problem solving expertise in a specific field. It was also realized that this expertise comes from diverse forms of knowledge accumulated by experts over a period of time. This has resulted in a knowledge-based approach to AI research. This approach has two basic tenets:

(i) Knowledge is power.

(ii) Knowledge is a precious resource.

Consequently, key research issues in this approach are acquisition, representation, and utilization of diverse forms of knowledge.

Knowledge Engineering is a subfield of AI that is concerned with the issues related to the representation and manipulation of the real world knowledge in a machine so that it can solve complex problems in a narrow field of expertise. *It is the art of tinkering with knowledge to make products useful to mankind*. Before proceeding to the applications to process synthesis, it is essential to review some of the basics of Knowledge Engineering.

There are two important characteristics of knowledge.

(i) Awareness: The machine should be aware of its knowledge; i.e., *it should know what it knows.*

(ii) Independence: The knowledge must be separate (independent) form the control mechanism that manipulates it.

From both these standpoints, a conventional program, written in a language such as BASIC, FORTRAN or Pascal, can not be said to possess knowledge. Knowledge can be expressed in two forms: declarative, or *knowing "that"* and procedural, or *knowing 'how'*. No single form is adequate to represent all kinds of knowledge. In fact, some domains require both forms to effectively represent their knowledge.

Over the last decade, the researchers in AI have proposed numerous schemes for knowledge representation (KR schemes). These include Logic, Production Rules, Semantic Nets, Frames, Conceptual Graphs, Functional Programming, and Object Oriented Programming (see, e. g., Barr and Feigenbaum, 1981; Rich, 1983; Sowa, 1984; Stefik and Bobrow, 1986). Each of these schemes have merits and demerits. Consequently, the choice of a scheme for an application depends upon the nature of the domain knowledge. Nevertheless, knowledge engineers, based on their experience, have realized the limitations of resorting to any one single scheme. Therefore, more powerful schemes, called hybrid systems, have been developed. These systems incorporate more than one individual scheme. Notable among such systems are LOOPS, KEE and ART. Such systems allow various parts of domain knowledge to be represented in different forms that are most appropriate.

Knowledge can be represented at a variety of levels. In a descending order, these levels are

(i) Linguistic
(ii) Conceptual
(iii) Epistemological
(iv) Logical
(v) Physical

Each of the KR scheme mentioned above is capable of representing the knowledge at various levels. Domain experts usually express their knowledge in linguistic form. Although this knowledge may be in a highly formal and systematized form, it still must be transformed at the level of the representation scheme. Any such transformation usually results in some degree of loss of knowledge; the lower the level, the greater the loss. Consequently, we must aim to represent our knowledge at the highest possible level. However, as long as we do not have machines that understand natural languages such as English, we have to settle for, at most, the conceptual level. Developing schemes to represent knowledge at the conceptual level is an area that is currently at the forefront of AI research.

KNOWLEDGE ENGINEERING FOR COMPUTER-AIDED PROCESS SYNTHESIS

Having seen the necessity of incorporating domain knowledge into computer programs to create "better" design/synthesis systems, some of the important issues of Knowledge Engineering need be examined prior to embarking on building such systems.

How much knowledge is enough?

This depends upon the nature of the problem domain. There are two extremes, game playing and newspaper story understanding. In game playing, the knowledge is important only to constrain the search for a solution whereas in (newspaper) story understanding, a large amount of knowledge is required even to be able to recognize a solution. With "unlimited" computing power, very little knowledge is needed to create a "perfect" solution for the problems of first type. In contrast, for the problems of second type, even with a great amount of knowledge, we can not guarantee a "perfect" solution. In process synthesis, Heat Exchanger Network (HEN) synthesis problem falls under the first type, whereas the problem of synthesizing entire process flowsheets falls under the second type.

At what level should the knowledge be represented?

The production rule formalism, by far the most "popular" KR scheme in numerous areas of application, represents knowledge at most at an epistemological level. On the other hand, a significant part of synthesis knowledge exists at a conceptual level. Also, the task of synthesis/design, as carried out by the experienced process engineers, involves conceptual manipulation of knowledge; each word or symbol used in our heuristic reasoning has an associated concept influencing the decision-making to a significant extent. Although it is desirable to choose as high a representation level as possible, some parts of knowledge may exist at lower levels. Consequently, we should have available a spectrum of levels, from conceptual to logical, and represent "chunks" of knowledge at suitable levels. The hybrid systems mentioned earlier, enable us to accomplish this.

Desired characteristics of resultant systems.

As is the case with any design procedure, before we attempt to "build" knowledge-based systems, we aught to know what features such a system should exhibit. The following are some of the important characteristics that can be expected from a synthesis system.

(i) The system should be smart enough to solve simple problems easily and to know when it is "stumped".

(ii) The system should be able to explain/justify its results and decisions.

(iii) The system should be able to judge the reliability of its own conclusions.

(iv) The system should be able to communicate smoothly with the users (clients) as well as with the domain experts.

(v) The system should be able to reason on various levels, resorting to different tools such as rules of thumb, mathematical models, and if necessary, detailed simulation.

(vi) The system should be capable of knowledge acquisition and updating by itself and/or by knowledge engineers in conjunction with the domain experts. It may not be entirely feasible to attain this feature in the immediate future, but the rapidly advancing AI technology should enable us to achieve this in a few years' time.

Having discussed some knowledge engineering issues pertinent to process synthesis, we are now ready to tackle the synthesis problems.

PROCESS SYNTHESIS PROBLEMS

Synthesis of a process flowsheet is usually carried out in various phases: selection of process route, selection and sequencing of unit operations, separation system synthesis, energy integration, and others. Associated with each of these phases is a synthesis subproblem. The following classification of synthesis problems is widely accepted (see, e.g., Westerberg, 1980; Nishida et al., 1981).

(a) Reaction path synthesis

Find a sequence of reactions which will lead to a given target molecule, starting with the available (or specified) raw materials.

(b) Separation systems synthesis

Synthesize a cost and/or energy minimizing sequence of separation systems that can isolate the specified products from the feed stream with known conditions such as composition, temperature, pressure, etc.

(c) Heat exchanger network synthesis

Find a cost and/or energy minimizing network of heat exchangers to meet the required condition of target temperatures of process streams by heating or cooling.

(d) Control system synthesis

Determine the control structure for a given process by selecting the controlled and manipulated variables and pairing them, so as to satisfy the control objectives.

(e) Entire process flowsheet synthesis

Develop a process configuration which converts the available raw materials to the desired products in the most economical fashion from the cost and/or energetic point of view.

Additional subgroups are being created as progress is being made in the already existing subgroups, e.g., energy transfer network synthesis involving mechanical work exchange as well as heat exchange to transfer energy, energy integrated separation sequence synthesis (currently restricted to heat integrated distillation sequences), and reactor network synthesis.

The solution to each of the problems, described in the preceeding paragraph, involves different kinds and amounts of knowledge. Furthermore, the phases of process synthesis are by no means independent or sequential; we need to iterate through the phases to obtain acceptable results. Thus building an automated synthesis system in its entirety is a task of considerable complexity; it requires a thorough understanding of the fundamentals of AI, concepts in Knowledge Engineering and expertise in process synthesis. It would be wise to solve each of the subproblems (a) through (d) in isolation. Once sufficient experience is gained by solving these subproblems, it would be easier to build a system to solve the entire problem by utilizing the resultant insights in the intricacies of the issues of conceptualization and implementation.

The problem of synthesizing a heat exchanger network is chosen for this work; it is the easiest and the most widely studied synthesis subproblem. As shall be seen in the next chapter, sufficient knowledge exists to conceive optimal or near optimal heat exchanger networks (HEN's) for a given problem. However, all of the currently available softwares have been developed using the conventional computerization approach, and they fall short of matching the performance of the experts. It would be interesting to see if a knowledge-based approach can improve upon this.

The following three-step strategy is adopted in this work:

(i) Knowledge extraction and formalization.

(ii) Development of schemes to represent and manipulate the knowledge.

(iii) Development of a conceptual design of a knowledge-based system for HEN synthesis.

Note that the scope of this work is limited only to the conceptual design; physical design (implementation) is left for the future. The outcome of this work will be a methodology/tool for HEN synthesis, which can be used and mastered by men or machines.

CHAPTER 3. A REVIEW OF HEAT EXCHANGER NETWORK SYNTHESIS

It is widely believed that the problem of systematic synthesis of heat exchanger networks (HEN's) was first introduced by Masso and Rudd (1969), although some previous work can be found in the literature, e.g., Whistler (1948) and Hwa (1965). Solution of this particular synthesis problem has progressed substantially since then. While scores of methods have since been reported, no single method has proven to be the best candidate for automatic generation of optimal networks; that there is a need for such a method is beyond any doubt. Some methods proposed in the recent past have made significant contributions towards fulfilling this need. The method proposed in this work exhibits some of the concepts from the selected methods of the past; thus a brief review of these methods is highly desirable for identifying the strength and usefulness of the proposed method.

This chapter begins with an overview of the work accomplished to date in the field of HEN synthesis. The problem specifications and the solution strategy are discussed first, followed by the detailed discussion of the three steps of HEN synthesis: preanalysis, network invention, and evolutionary modifications. Finally, the representation scheme for the solution adopted for this work, is described.

PROBLEM SPECIFICATIONS

The heat exchanger Network synthesis problem can be formulated as follows:

> Given a set of process streams, with specified flow rates and heat capacities, find the energy-optimum and minimum-cost set of heat transfer units (HTU's) that will transform the given initial (source) temperatures of all the streams to the desired final (target) temperatures.

The following simplifying assumptions are usually made (see, e.g., Nishida <u>et</u> <u>al</u>., 1981).

(a) The utility streams, such as steam and cooling water, are available at desired temperatures. The flow rates are not specified, but are assumed to be as much as needed by the problem.

(b) The temperature effect on heat capacities can be ignored.

(c) Each HTU is either a counter-current single-pass heat exchanger, a heater, or a cooler.

(d) The necessary cost data, viz. the correlation for the investment (capital) cost of an HTU as a function of its area, and the annual cost per unit flow for each of the utility streams are available.

(e) The heat transfer coefficients, which may be stream/stream match dependent, are available for all the HTU's.

Under these assumptions, the specifications of the HEN synthesis problem correspond to the information typically available from a process flowsheet which is yet to be heat integrated but for which the heat and material balances have been performed. Thus, it represents a reasonably realistic and useful problem.

Even for a relatively small problem, for which the number of required HTU's is small, the number of possible alternative HEN configurations is substantially large (see, e.g., Motard and Westerberg, 1978). The synthesis problem is to find a network that has the least annualized cost. This annualized cost includes the investment (capital) cost for the HTU's, converted to a cost per year basis, and the annual cost of the necessary utilities. The problem becomes that of trading the capital cost against the utility cost.

Almost all the earlier methods require computation of the total annual cost in determining the "best" solution. This, in general, gives rise to a great deal of inefficiency since, without any approximation, the cost computations are rather involved; in contrast, computations needed for the network generation are much simpler. Furthermore, the changes in the component costs may necessitate repeating the entire solution procedure. A little insight to the problem enables us to circumvent this predicament by conceiving an alternate basis of evaluating a network that is independent of the cost. The investment (capital) cost depends largely on the number of HTU's and their heat loads. For a given total heat load, the networks with the least number of units obviously will have the lowest costs. The operating cost depends upon the quantities of utilities consumed; in most cases, it is mainly the quantity of hot utility (usually steam). The utility costs form an overwhelmingly large share of the total cost; consequently, the following selection rule can be stated:

> *Among various candidates, the best or the cost-minimizing network must have the lowest utility requirements and the fewest number of HTU's, in that order.*

Naturally, there can be more than one networks satisfying this criterion. In such cases, the selection can be made based on additional criteria. Adopting this criterion eliminates the need for computing the costs of the candidate networks and hence eliminates the need for assumptions (d) and (e) stated at the begining of this section. Also, it enables a method to disregard the precise knowledge of the costs and the heat transfer coefficients; both can be altered without affecting the network structure and without sacrifising the optimality of the chosen solution. This rule will be assumed throughout the course of this work. As a consequence, any method that does not resort to this rule, i.e., any method that involves cost calculations, will not be explored subsequently.

The problem has now been transformed from the cost domain to the domain of structural characteristics of the network. Interesting foresights can be obtained directly from the problem specifications regarding the optimality of the network that we attempt to generate. Three major results, pertaining to the three important properties of an optimal network, are available. Two of these were stated by Hohmann (1971), and the third one was hinted at by him, but not explored. The first two of these results are that we can predict *a minimum utility* *usage target* and *the fewest possible number of HTU's* required for a problem, prior to developing an actual network structure. In other words, these two aspects are independent of the network structure. These targets can *usually* be met in an actual design, which then turn out to be the most economic one. Linhoff and his co-workers have systematized these two results (Boland and Linhoff, 1978; Linhoff and Flower, 1978a; Linhoff, 1979). Also, the conjecture by Hohmann that both the targets could always be met, if stream splitting is permitted, has been shown by Linhoff not to be true by a counterexample (see, e.g., Linhoff et al., 1981). The third major result is that we can locate the "bottlenecks" called pinches in a process design, again prior to developing a network solution. The process designs can be altered or revised through the discovery of these bottlenecks or pinches, which preclude further heat integration (see, e.g., Linhoff, 1979; Umeda et al., 1979a; Umeda et al., 1979b). These results are industrially significant and would by themselves justify the research expended to date in the entire area of process synthesis (Nishida et al., 1981).

The solution to a HEN problem can be partitioned into the following three major steps (Nishida et al., 1981).

1. Preanalysis to set the targets for the optimal network.
2. Network Invention to conceive an "initial" network.
3. Evolutionary modifications of the "initial" network.

Almost all the HEN solution methods proposed so far can be broken into these steps; some combine the last two steps. The succeeding three sections discuss these steps in detail.

PREANALYSIS

Preanalysis involves establishing the targets for the network to be designed. Recapitulating from the preceeding section, these targets are the minimum utility and the minimum number of HTU's required to solve the problem. In addition, some of the earlier methods involved prediction of the minimum total area required for the network. Since we have already transformed the problem from the cost domain to the domain of the structural characteristics, we are no longer dealing with the cost of the resultant HEN; therefore, the area calculations are not required.

The minimum utility requirement target can be arrived at in the following five ways (Nishida et al., 1981).

UB-1. Net difference between the heating needed for the cold streams and the cooling needed for the hot streams.

UB-2. Same as UB-1, but modified to account for the portions of hot streams that are colder than all of the cold streams, and the portions of cold streams that are hotter than all of the hot streams.

UB-3. Exact bounds accounting for a uniform minimum allowed approach temperature.

UB-4. Same as UB-3, but modified to account for the user defined stream/stream match restrictions.

UB-5. Same as UB-4, but with stream/stream match dependent minimum approach temperatures.

UB-1 and UB-2 were used by some of the earlier workers (see, e.g., Rathore and Powers, 1975; Grossman and Sargent, 1978). These bounds are not always feasible, since they do not take into consideration the restriction imposed by the second law of thermodynamics, viz., the hot stream must be hotter than the cold stream with which it is matched. Presence of a pinch point, except when at one of the ends, will guarantee the failure (infeasibility) of these bounds.

UB-3 gives the exact bounds which can always be met, since it accounts for the second law restriction by imposing a minimum allowable approach temperature for a match to be feasible. This method bas been by far the most "popular" one (see, e.g., Hohmann, 1971; Linhoff and Flower, 1978a; Umeda _et al_., 1978; Flower and Linhoff, 1979; Greenkorn _et al_., 1980). UB-4 and UB-5 were first proposed and used by Cerda _et al_. (1980). These bounds allow the users to exclude matches between designated pairs of streams, either in total or over a certain temperature range. Also, UB-5 permits the minimum approach temperature to be stream/stream match dependent. The problem is formulated as a "network flow" or an assignment problem in linear programming.

As for the second target, that of predicting the minimum number of HTU's, the rule proposed by Hohmann (1971) is still widely used: *the minimum number of HTU's for a HEN is usually, but not always, one less than the total number of streams, including the utility streams.* Thus,

$$u_{min} = N - 1 \tag{3-1}$$

where u_{min} is the (probable) fewest number of HTU's required for a problem, and N is the total number of streams, including the utility streams. Linhoff _et al_. (1982), in an attempt to amplify Hohmann's result, studied the conditions under which Eq. 3-1 does not hold. They have proposed the following modified result.

$$u_{min} = N + L - s \qquad (3\text{-}2)$$

where u_{min} and N are the minimum number of HTU's and total number of streams, respectively (same as in Eq. 3-1), L is the number of loops (cyclic structures where the same two streams are matched more than once, with other matches in between), and s is the number of subsets of streams that form independent subnetworks. However they have not explicitly stated the effect of stream splitting on Eq. 3-2.

Linhoff et al. (1982) also noted that it is not always possible to simultaneously attain both the targets, even if the stream splitting is permitted. Through analysis of all such cases, they have discovered the concept of pinch point in a HEN. [This was the third result hinted at by Hohmann (1971).] A pinch point is a temperature where the "bottleneck" in heat integration occurs. Any HEN problem always has at least one pinch point and very rarely does it have more than one (although it can not be ruled out). If the pinch point is at one of the ends of the network (i.e., either the highest of the hot stream source temperatures or the lowest of the cold stream source temperatures), then Eq. 3-2 does indeed predict the minimum number of HTU's that will constitute the HEN's featuring the minimum utility consumption. However, if the pinch point is between these two extremes, then the actual minimum number of HTU's required to generate a HEN with minimum utility consumption is more than the number obtained from Eq. 3-2 (cf. Chapter 4). Once again, this analysis does not hold entirely when one or more streams are split during the network generation step (cf.

chapter 5). The concept of pinch point is extremely important for HEN synthesis because, for a HEN to have minimum utility consumption, no heat transfer should take place across the pinch point (see, e.g., Linhoff and Hindmarsh, 1982; Linhoff _et al_., 1982). Also, the problem is most constrained at the pinch point since the driving force for the heat transfer is minimum at the pinch. Pinch points are independent of the HEN configuration; they are characteristic of only the problem specifications. As a consequence, locating the pinch points must be the first step in the preanalysis and hence in any HEN synthesis method.

NETWORK INVENTION

The network invention step can be divided into four substeps (Nishida et al., 1981).

(a) Partitioning the problem.

(b) Merging equivalent heat sources and sinks.

(c) Selecting stream/stream matches (i.e., which streams to match).

(d) Selecting the heat loads of the HTU's to produce feasible matches between the selected streams.

Note that all the substeps are not necessarily present in all of the methods proposed so far; some combine two or more of these steps whereas some others omit a step or two.

Partitioning, if present in a method, is usually done in one of the two ways: by some key temperature intervals or by pinch points. Some methods merge the streams with equivalent heat (within the same interval) to form hot and cold superstreams. Merging of streams is characteristic of minimum area algorithms. More often than not, the last two steps, (c) and (d), are combined together. Once the decision regarding which streams to match is made, selecting the HTU size or load to generate a feasible match is fairly straight forward; to ensure that the HEN can attain the target of the minimum number of HTU's, we must maximize the quantity of heat transferred in each unit. The second law of thermodynamics determines the extent of feasible matching: the driving force for the heat transfer must remain above zero. Thus, we should choose the magnitude of heat duty for an HTU such that either at least one of the streams is "eliminated" from the problem for any further consideration (i.e., its heat load is satisfied) or the driving force for the heat transfer reduces to the minimum allowable value, since it is not possible to attain a zero driving force. Naturally, the selection of HTU load depends upon which streams are being matched. Consequently, it is of utmost importance that we make the "right" choices in selecting the streams to be matched. Although this selection has been accomplished in many ways, two broad classes can be defined. Some methods carry out this step as a sequence of match decisions, while others make their decisions in parallel.

Sequential Match Decision Algorithms

The essence of these algorithms developing a HEN as a sequence of match decisions, is to construct a tree of networks, with the initial node (the root of the tree) being the network with no matches, i.e., the original problem. The children of this node are all the networks containing one (feasible) match, their children contain two (feasible) matches and so on; each child contains one more match than its parent. Each leaf node of the tree is a network containing all the matches (i.e., a fully heat integrated network).

Algorithms of this type have two steps: generation of the sequential match tree and selection of a leaf node that best satisfies the optimality criteria. One of the earlier algorithms of this type, proposed by Pho and Lapidus (1973), suggests developing the entire tree of networks in accordance with their rules; however, the tree becomes excessively large even for a reasonable size problem. More recent algorithms of this type use an improved approach, generating only one level of nodes at a time. One of the node is selected from amongst all the children, and the nodes of the next level are generated for this selected node only. A number of tree search methods are available, including the depth first, breadth first, branch and bound, and heuristic methods, which combine the tree generation and search. To reduce the search space, several criteria for match restrictions are employed which prevent, directly or indirectly, the violation of the optimality constraints established during the preanalysis step. Various match restrictions incorporated in different methods are listed below (see, e.g., Nishida _et_ _al_., 1981).

MR-1. Disallow stream splitting.

MR-2. Disallow stream/stream rematching (to prevent cyclic structures).

MR-3. Disallow a match if it precludes the predicted minimum utility usage.

MR-4. Disallow a match if it precludes a network having the predicted fewest number of HTU's.

The methods based on the branch and bound approach require extensive computations, but they tend to guarantee the globally optimal solution. Hence this approach has been quite popular (see, e.g., McGalliard, 1971; Rathore and Powers, 1975; Greenkorn _et_ _al_., 1978; Grossmann and Sargent, 1978). Heuristic methods, on the other hand, do not guarantee an optimal (local or global) solution, but they require relatively little computational effort. The optimality of the solution depends upon the set of heuristics and order of their application.

Summarized below are some of the common heuristics employed for selecting a pair of streams to match (Nishida et al., 1981).

HR-1. Select the hot stream with the highest source temperature and the cold stream with the highest target temperature.

HR-2. Select the hot stream with the coldest target temperature and the cold stream with the coldest source temperature.

HR-3. Select the match giving the least value of average temperature difference.

HR-4. Select the match giving the least value of the estimated upper bound on the overall network cost.

Upon selection of the streams to be matched, decisions regarding which portions of heat to be transferred need be made. One (each for the hot and the cold streams) of the following heuristics, called stream heat selection rules, is employed for this purpose (Nishida et al., 1981).

HS-1*h*(*c*). Take heat from (supply heat to) the hottest end of a hot (cold) stream.

HS-2*h*(*c*). Take heat from (supply heat to) the coldest end of a hot (cold) stream.

HS-3*h*(*c*). Take heat from (supply heat to) the intermediate portion of a hot (cold) stream.

The heat selection options (HS-1*h* to HS-3*h* and HS-1*c* to HS-3*c*) can have profound effect on the resultant network. Earlier methods match the hottest portion of the hot stream against the coldest portion of the cold stream (HS-1*h*/HS-2*c*), leaving the hot end of the cold stream and/or the cold end of the hot stream for subsequent heating or cooling, most probably by utility streams. Of course, for the current match, this will give rise to an HTU with the minimum area, but the final result will be a network with a greater number of HTU's and perhaps, higher utility requirements. This is because it is quite likely that the hot end of the cold stream will be at a higher temperature than the hot end of the rest of the hot streams, thus prohibiting further heat exchange and consequently requiring "extra" amount of utilities.

Ponton and Donaldson (1974) were the first to advocate the matching of the hot end of a hot stream with the hottest portion of the cold stream (HS-1*h*/HS-2*c*). This heat selection rule is quite reasonable for the above-ambient networks, as it tends to allow the lower temperature hot utility to be used (if such a selection is available), in the process needing the cold utility at a lower temperature. This trade-off is reasonable since the above-ambient hot utilities are far more expensive than the above-ambient cold utilities.

Some of the more recent papers describe the sequential algorithms simultaneously satisfying the constraints of minimum utility usage (MR-3) and using the (predicted) minimum number of HTU's (MR-4) (see, e.g., Greenkorn et al., 1978; Grimes et al., 1980).

Simultaneous Match Decision Algorithms

To establish the stream/stream match decisions "in parallel" or "simultaneously", two approaches have been used by the methods published so far (Nishida et al., 1981). The first one was employed by some of the earliest workers and is based on the ***assignment problem*** in linear programming. According to this approach, each of the streams, including the utility streams, is partitioned into a set of smaller substreams having identical heat contents. The partitioning can be sequential (see, e.g., Kesler and Parker, 1969; Cena et al., 1977), which is equivalent to no stream splitting, or parallel (see, e.g., Kobayashi et al., 1971), which is equivalent to splitting the streams. The problem is to assign the hot substreams to the cold ones in a manner which minimizes the sum of the cost associated with each assignment; constraints precluding certain assignments can be readily added. The basic constraints arise out of thermodynamic considerations, with the additional user defined constraints being the optional ones. The objective function is the total cost of the network. This approach has rarely been used in recent years because it involves cost calculations and generates a large number of substreams and constraints, even for a relatively small problem (Nishida et al., 1981).

The second simultaneous match decision approach is the thermodynamic-combinatorial (TC) method (Linhoff, 1979; Flower and Linhoff, 1980). This method generates all the HEN's satisfying the match restrictions MR-1 (no stream splitting), MR-3 (minimum utility usage), and MR-4 (predicted fewest number of HTU's). In addition, the resultant HEN's are acyclic. The method involves little computational effort, but may terminate without yielding any solution. This second approach to simultaneous match decision is strikingly different from that of the rest of the HEN synthesis methods. The approach can be used effectively to yield a powerful, yet simple method for generating optimal HEN's.

Selection of Network

For a sequential match decision algorithm, this step is accomplished at the same time when the stream/stream match decisions are made. In contrast, it is carried out as a separate step for a simultaneous match decision algorithm. The methods which merge equivalent heat contents (e.g., Nishida et al., 1971) have only selected which merged heat sources are to supply heat to which merged heat sinks. It is left to the user to choose among the various possible alternatives, which pairs of streams to match for developing a final network. In the thermodynamic-combinatorial (TC) method, all the networks having positive approach temperatures (for all the matches) are retained.

EVOLUTIONARY MODIFICATIONS

Most early methods do not include this step; each method attempts to generate the optimal network at one shot -- a hit or miss approach. McGalliard and Westerberg (1972) were the first to advocate the evolutionary approach; the approach allows a non-optimal HEN to be generated as the first step of the method, which is then updated subsequently. They have proposed a method to determine if a modification to a flowsheet leads to an improved flowsheet without requiring either the original or the modified flowsheet to be optimized. The method is based upon primal/dual bounding. Since then many methods have been proposed to improvise (non-optimal) HEN's. Shah and Westerberg (1975) have presented a set of evolutionary rules for carrying out "small" modifications in a HEN in a recursive fashion. The crucial problem is to access what type of changes constitute "small modifications". Nishida et al. (1977) have proposed a set of evolutionary rules to improve networks developed by the minimum area algorithm of Nishida and coworkers (Nishida et al., 1971), which gives rise to HEN's containing an excessive number of HTU's. These evolutionary rules were to reduce the number of HTU's.

The evolutionary development (ED) method of Linhoff and Flower (1978b) consists of a set of rules for reducing the number of HTU's and the total heat transfer area of a HEN. The method was primarily devised for improving the HEN's generated by their TI method (Linhoff and Flower, 1978a), although it can be used to improve any non-optimal network. The rules allow only those modifications which lead to a thermodynamically feasible network. While suggesting that the method is designed to reduce the number of HTU's to a minimum, it does not give any specific strategy for utilizing the rules to achieve that goal. The thesis by Grimes (1980) proposes two evolutionary rules to find the neighbouring structures of the original network, all satisfying the match restrictions MR-3 (minimum utilities) and MR-4 (predicted fewest HTU's). Furthermore, he has presented theorems showing that all such structures can be reached from any starting structure through successive evolutionary steps by resorting only to his two rules. Su (1979) has proposed an evolutionary loop-breaking algorithm for optimizing the initial networks generated by the TI method of Linhoff and Flower (1978a). The method also explores the possibility of stream splitting to achieve its goal. The same method has been successfully used by Lin (1983) to improve the network generated by the pinch design (PD) method of Linhoff and Hindmarsh (1982). As can be seen, in spite of decoupling the two steps of network generation and evolution, the evolution methods are still closely related to the manner in which the initial network is generated.

PROBLEM REPRESENTATION

Several representations have been used in developing HEN's.

(a) Temperature enthalpy diagram (Whistler, 1948).

(b) Simple match matrix (Pho and Lapidus, 1973).

(c) Heat Content diagram (Nishida et al., 1971).

(d) Grid representation (Linhoff and Flower, 1978a).

An excellent discussion of all these schemes can be found in the review papers by Westerberg (1980) and Nishida et al. (1981). The first three schemes do not permit the network representation; as a result, separate networks must be drawn. In contrast, the last scheme incorporates the network representation, so that the networks are easily visible, even the partial ones obtained during the process of synthesis. Also, this method of representation is most suitable for evolutionary modifications. Hence, the grid representation scheme is chosen for the present work. Before we proceed any further, a brief description of the scheme is in order.

Figures 3-1 and 3-2 show typical grid diagrams. Each stream is represented by a directed line, going from the source to the target temperature both of which are labeled at each end of the stream. All the hot streams are drawn at the top with the source temperatures on the left-hand side and the target temperatures on the right-hand side, i.e. the hot streams "go" from the left to the right at the top of the diagram. The cold streams are drawn at the bottom of the diagram with the source temperatures on the right-hand side and the target temperatures on the left-hand side; i.e., the cold streams "go" from

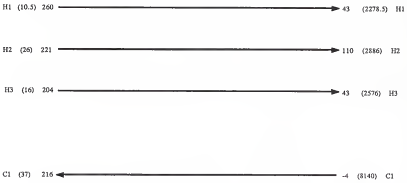


Fig. 3-1. A typical grid diagram for a HEN problem.

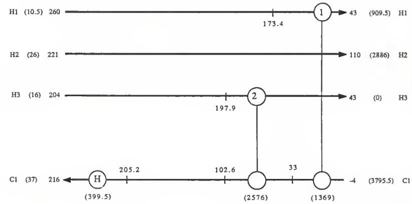


Fig. 3-2. Grid diagram for a partial solution to the HEN problem in Fig. 3-1.

right to left at the bottom of the diagram. Note that for all the streams, hot and cold, the higher temperatures are at the left end of the diagram. Two values are associated with each stream; on the left edge of the stream is the heat capacity flow rate (tbe product of flow rate and specific heat of the stream), and on the right is the *unsatisfied* heat load of the stream. To distinguish these values from the source and the target temperatures, they are parenthesised. Finally, the identification tags of the streams, e.g., H1, H2, ..., and C1, C2, ... etc., are displayed at both the ends of the streams. With this description, we can now represent any HEN problem into the grid diagram form. Nevertheless, to represent the solutions (full and partial), additional information is required (see Figure 3-2).

Each HTU is represented by a circle on the corresponding stream(s). A heater and a cooler has H and C, respectively, within the circle. A heat exchanger involving two streams is represented by a vertical line connecting the two circles on the corresponding streams. The top circle (of a heat exchanger) contains the identification number indicating the sequence in which the heat exchanger has been created. Thus the heat exchangers in a HEN will be labeled 1, 2, 3, and so on. The heat duty of a heater is displayed in parenthesis below the corresponding circle, that for a cooler in displayed above the corresponding circle, and for a heat exchanger, it is displayed below the "bottom" circle, i.e., the one on the cold stream. Intermediate temperatures of the streams, if needed, are displayed above the position for cold streams and below the position for the hot streams; the position is indicated by a small vertical bar at the appropriate location on the stream (see, Figure

3-2). The temperatures on all streams decrease from the right to the left, though not to a scale. Armed with these conventions we can now represent any HEN in a grid format without any ambiguity. The units for the values are not shown anywhere in the diagram; there are no restrictions except that all the values must be in a consistent set of units. Unless otherwise mentioned, the standard set of units will be used throughout this work: Temperatures in °C, heat loads in kcal/hr and heat capacity flow rates in kcal/hr-°C.

In addition to the grid diagram, the temperature-enthalpy diagrams are needed to gain insights into some of the concepts (see, e.g., Nishida et al., 1981, Westerberg, 1980). As shown in Figure 3-3, a stream in such a diagram is represented by a line in a two-dimensional coordinate system with the enthalpy on relative basis, denoted by H, as the X-axis and the temperature, denoted by T, as the Y-axis. Thus, a hot stream goes from the top-left to the bottom-right, and a cold stream in exactly opposite direction. The same diagram can be used to represent sets of streams with x-axis now representing the cumulative enthalpy. Note that enthalpy scale is only relative; thus, streams may be moved to the right or to the left without any effect. A match between the two streams is represented by placing a cold stream directly below a hot stream. Where the streams overlap, the match takes place. A match is thermodynamically feasible if and only if the hot stream is "above" (hotter than) the cold stream in the entire span along the match. The vertical distance between the streams is the temperature difference (driving force) experienced along the match. Note also that the left-hand edge of any stream is at a higher temperature than the

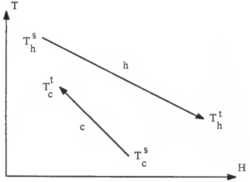


Fig. 3-3. A typical temperature - enthalpy diagram.

right-hand edge. Consequently, for any match, the left edge will be referred to as the hot end and the right edge as the cold end. Additionally, the slope is inversely proportional to the heat capacity flow rate of a stream; thus a steeper stream (higher slope) has a lower heat capacity flow rate whereas a shallower stream (lower slope) has a higher heat capacity flow rate. Once again, the diagram is not necessarily drawn to any exact scale.

The grid diagram will be used to show the solutions (full and partial) of the HEN problem while discussing and demonstrating the proposed solution strategy. On the other hand, the temperature-enthalpy diagram will be used to demonstrate the relevant "theoretical" issues during the knowledge extraction and formalization phase.

CHAPTER 4. MINIMUM NUMBER OF UNITS FOR HEAT EXCHANGER NETWORKS

As discussed in the preceeding chapter, two targets are established during preanalysis: the minimum utility consumption and the minimum number of HTU's. A variety of methods and algorithms are available to predict the minimum utility consumption (cf. Chapter 3). However, none of the relationships proposed so far in the literature enables us to predict the "exact" or guaranteed minimum number of HTU's for a HEN featuring the minimum utility consumption. In fact, Linhoff et al. (1982) have reported that it is not always possible to simultaneously attain both targets, even if the stream splitting is permitted. However, they assume that the minimum number of units is given by the relationship proposed by them (cf. Eq. 3-2). Thus, according to their analysis, the optimality with respect to the number of HTU's of a HEN, generated by any method, can not be guaranteed unless an exhaustive search is performed over the entire solution space. This obviously defeats the purpose of predicting or setting a target; if there is a possibility that the actual number of HTU's may be higher or lower than the predicted target, then the importance of setting the target will diminish drastically. To prevent such a situation, a means is needed for predicting an *always attainable* target (of minimum number of HTU's) which will never be exceeded. To accomplish this, we should search for the cause that gives rise to the minimum number of units. This is in accord with the intent of the knowledge-based approach, which seeks to formalize the underlying knowledge behind any cause-effect relationship.

In this chapter, we shall examine the cause-effect relationship for setting the target of the minimum number of HTU's for a HEN.

ELIMINATION STRATEGY FOR MINIMUM NUMBER OF UNITS

Each HTU in a HEN is a result of matching a pair of hot and cold streams. Whenever such a match is made, the heat duties of both streams are satisfied fully or partially, thus effectively reducing the problem. The outcome of each match is one of the following three possibilities;

- (a) The heat duties of both streams are fulfilled, thereby eliminating them from any further consideration.
- (b) The heat duty of one of the two streams is fulfilled, thereby eliminating it from further consideration.
- (c) The heat duty of neither of the two streams is fulfilled, thus both remain under consideration for subsequent matching.

Depending upon the outcome of the match, we shall classify it to be of type (a), (b) or (c).

The ultimate goal of HEN synthesis is to "eliminate" all streams from consideration by fulfilling their heat duties. An ideal situation, therefore, would result if both (hot and cold) streams can be eliminated in all matches, i.e., all matches are of type (a). Such a situation occurs rarely in practice; it is extremely unlikely that a HEN problem will have all its streams as *conjugate pairs* of hot and cold streams such that they have exactly identical heat duties and thermodynamically compatible source and target temperatures. The next best situation would be to have all the matches of either type (a) or type (b), i.e., each match eliminating at least one stream. As we shall see in subsequent sections, it is indeed possible to generate HEN solutions by following this strategy. Making a match such that no stream gets eliminated [type (c) match] will give rise to a HEN having more than the minimum number of HTU's. This is because to eliminate any stream, at least one match is required. The "remaining portions" of the streams from the "incomplete" [type (c)] match will therefore require two additional matches; thus at least three matches will be required for two streams. However, if we restrict the solution method to create matches of types (a) and (b) only, we can "eliminate" two streams in two matches. In summary, the following guideline can be proposed to attain the minimum number of HTU's for a HEN.

> *To generate a HEN featuring the minimum number of HTU's, let each match "eliminate" at least one of the two streams and if possible, both.*

Any method adhering to this guideline will always generate HEN's featuring the fewest number of HTU's. Note that while this guideline ensures the attainment of the fewest number of HTU's, it does not yield this number apriori. Consequently, to assure the optimality of the generated HEN, all possibilities of making matches of type (a) must be explored during the synthesis. However, if we can predict the number of matches of type (a) and (c) for a HEN, it is possible to predict the "exact" minimum number of HTU's in a HEN using the relationship derived in the succeeding paragraphs.

Let a, b, and c be the number of matches of types (a), (b), and (c), respectively, in a HEN. Then, the number of HTU's in this HEN, u, is given by

$$u = a + b + c \qquad (4\text{-}1)$$

Also, the total number of streams in the HEN including the utilities (N) is related to the matches as follows:

$$N = 2a + b \tag{4-2}$$

To minimize the number of HTU's in a HEN, the number of matches of type (a) must be maximized (to a_{max}) and the number of matches of type (c) must be minimized (to c_{min}). Thus, for minimizing the total number of HTU's, the number of matches of type (b) is given by, from Eq. 4-2,

$$b = N - 2a_{max} \tag{4-3}$$

Hence, the minimum number of HTU's (u_{min}) for the HEN is given by

$$\begin{aligned} u_{min} &= a_{max} + (N - 2a_{max}) + c_{min} \\ &= N - a_{max} + c_{min} \end{aligned} \tag{4-4}$$

Note that the minimum value of a_{max} is 1, since at least one match of type (a) is always present in a HEN. This is because the original problem (with utilities) is already "heat-balanced" so that the last match will always eliminate both, the hot and cold streams. Also, it is possible to prevent the matches of type (c), as discussed in the subsequent sections. Under these conditions, with only one match of type (a) and none of type (c), Eq. 4-4 reduces to

$$u_{min} = N - 1 \tag{4-5}$$

This relationship is identical to that proposed by Hohmann (1971), i.e., Eq. 3-1.

Equation 4-4 also explains the relationship proposed by Linhoff et al. (1982) to predict the minimum number of HTU's for a HEN, i.e., Eq. 3-2, which can be written as

$$u_{min} = N - s + L \tag{4-6}$$

Each match of type (a) partitions the network by creating an independent subset of streams (which form an independent network). On the other hand, each match of type (c) gives rise to one (closed) loop in the HEN, since the unfulfilled heat duties of the two streams must be matched with some streams from the rest of the network. Thus Eqs. 4-4 and 4-6 are exactly identical.

EFFECT OF PINCH POINT

As mentioned in the preceeding chapter, the effect of pinch point on the relationship predicting the minimum number of units (Eq. 4-6) has not been studied, even though the presence of pinch points precludes the attainment of minimum units target as predicted by Eq. 3-2. In this section we shall derive the exact relationship for predicting the minimum number of units in the presence of a pinch point. The target of the minimum number of units given by this relationship will always be attainable simultaneously with the other target, the minimum utility consumption.

The maximum energy recovery and hence the minimum utility consumption is possible if and only if (see, e.g., Linhoff et al., 1982)

(a) no heat is transferred across the pinch,

(b) no hot utility is used below the pinch, and

(c) no cold utility is used above the pinch.

Figure 4-1(a) depicts the temperature-enthalpy diagram for a typical HEN problem with a pinch point. The driving force at the pinch point is, by definition, the minimum allowable, ΔT_{min} (see, e.g., Linhoff and Hindmarsh, 1982; Linhoff et al., 1982). Q_h and Q_c are the hot and cold utility requirements, respectively. Figures 4-1(b) and 4-1(c) illustrate the heat balance for the HEN problem. Note that any amount of heat (ΔQ) transferred across the pinch necessitates an equivalent amount of additional hot and cold utilities. Consequently, no heat should be transferred across the pinch for the minimum utility consumption. Thus, in essence, the pinch divides a HEN into two

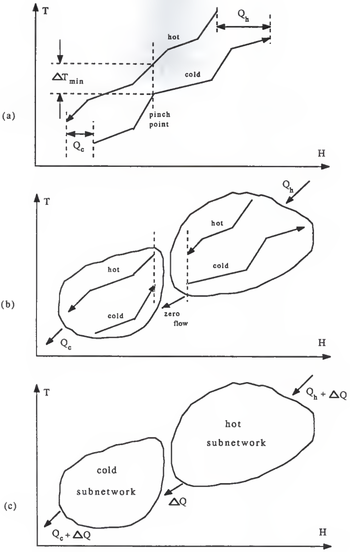


Fig. 4-1. Pinch decomposition (Linhoff et. al., 1982).

independent subnetworks, one above the pinch point (hot subnetwork) and the other below it (cold subnetwork). As seen from Figure 4-1(b), the pinched ends of both subnetworks, i.e., the cold end of the hot subnetwork and the hot end of the cold subnetwork, are perfectly "heat-balanced" (again, by definition of the pinch point). Consequently, no cold utility is required for the hot subnetwork, and no hot utility is required for the cold subnetwork. This explains the conditions (b) and (c); viz., no hot utility below the pinch and no cold utility above the pinch.

A HEN problem is said to have no pinch (called unpinched problem) when the pinch point lies at either end of the HEN. In such cases, the problem consists of only one subnetwork, and hence needs only one type of utility, either hot or cold, depending upon the end at which the pinch point is located.

Since no heat transfer is permitted across the pinch point, any stream in a subnetwork must be matched with another one from the same subnetwork. Thus, two separate networks need be synthesized; therefore, Eq. 4-4 holds for both the subnetworks, thereby giving rise to

$$u_{min}^{hot} = N^{hot} - a_{max}^{hot} + c_{min}^{hot} \tag{4-6}$$

and

$$u_{min}^{cold} = N^{cold} - a_{max}^{cold} + c_{min}^{cold} \tag{4-7}$$

Consequently, the minimum total number of HTU's in a *pinched problem* is

$$\begin{aligned} u_{min} &= u_{min}^{hot} + u_{min}^{cold} \\ &= (N^{hot} + N^{cold}) - (a_{max}^{hot} + a_{max}^{cold}) + (c_{min}^{hot} + c_{min}^{cold}) \end{aligned} \tag{4-8}$$

Note that $(N^{hot} + N^{cold})$ is not the same as the total number of streams N in the overall HEN problem, since there is at least one stream (and usually more) that straddles the pinch point, thus belonging to both the subnetworks. If N_p is the number of such streams, then we have

$$N^{hot} + N^{cold} = N + N_p \tag{4-9}$$

Based on this relationship, we have from Eq. 4-8,

$$u_{min} = (N + N_p) - a^{tot}_{max} + c^{tot}_{min} \tag{4-10}$$

where the superscript *tot* on the last two terms represent the summation of the corresponding terms (viz., a_{max} and c_{min}) for the constituent subnetworks (hot and cold). Note that the minimum value for a^{tot}_{max} is 2, since each subnetwork is heat-balanced and hence will have at least one match of type (a). Once again, type (c) matches can be prevented, thereby eliminating the last term in Eq. 4-10. Thus, for a pinched problem, if we generate a HEN featuring the minimum utility consumption exclusively by the matches of type (b), then the minimum number of HTU's are obtained by the following relationship;

$$u_{min} = N + N_p - 2 \tag{4-11}$$

Although this relationship has been derived for a pinched problem, it can be readily transformed into the corresponding relationship for an unpinched problem (Eq. 4-5) by assigning a value of 1 to N_p. This suggests that an unpinched problem is equivalent to a pinched problem with one stream straddling the pinch point (i.e., $N_p = 1$), as far as the setting the target of minimum number of units is concerned.

Equations 4-10 and 4-11 imply that when a HEN synthesis has to be carried out as a part of entire process synthesis; the decisions pertaining to the process synthesis should be made such that the number of streams straddling the pinch point (N_p) is minimized, if such a choice is available.

NECESSARY AND SUFFICIENT CONDITIONS FOR ELIMINATION

We have seen that to attain the minimum number of HTU's in synthesizing a HEN, we should match only those pairs of streams resulting in elimination of at least one of the streams from further consideration. This will ensure that the final network will not contain any match of type (c). We, therefore, need necessary and sufficient conditions under which this criterion will be satisfied. To arrive at such conditions, we should inquire: *"What prevents the elimination of at least one of the streams in a match?"*

Figure 4-2 illustrates a typical situation when a pair of streams are being selected for matching. The temperatures of the hot stream (h) are designated by T_h and those of the cold stream (c) by T_c, with superscripts s and t denoting the source and target temperatures. The remainder of the HEN problem, which may contain any combination of matched and unmatched streams, is represented by the rectangle labeled HEN. The left edge of this rectangle is termed as the hot end and the right edge as the cold end, since the temperatures at the left edge are higher than the corresponding ones on the right edge. The temperature difference between the two streams at the hot end is given by

$$\Delta T_{he} = T_h^s - T_c^t \tag{4-12}$$

and that at the cold end is given by

$$\Delta T_{ce} = T_h^t - T_c^s \tag{4-13}$$

We need to ensure that the match between the two streams will eliminate one of them (the one with the lower heat duty). Towards this end, is it

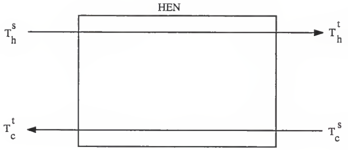


Fig. 4-2. Stream/stream match selection.

sufficient that the driving forces, ΔT_{he} and ΔT_{ce}, be greater than (or equal to) the minimum allowable driving force? As shown in the succeeding paragraphs, it is necessary, but not sufficient, that at least one of the two driving forces be greater than (or equal to) the minimum allowable driving force. The necessary and sufficient condition is that *at all points throughout the match, the driving force* ($T_h - T_c$) *must be greater than the minimum allowable value* (ΔT_{min}). If we are to resort to this condition for selecting the stream/stream match, it must be quantified in terms of the known values, viz., the heat capacity flow rates (mc_p values), the heat duties (Q values), and the stream temperatures (T_h and T_c values).

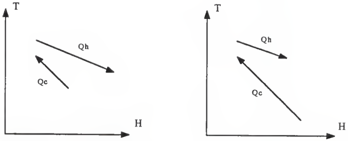
As shown in Figure 4-3, there are two ways in which a pair of streams can be matched; one way is to match them at the hot end (hot end match) and the other, to match them at the cold end (cold end match). For both types of matches, it is possible to eliminate one of the streams. We shall derive the conditions for each type of matches. Figure 4-4 shows the driving force at an intermediate point in a match, upto which Q units of heat has been transferred. This driving force is given by

$$\Delta T = T_h - T_c \tag{4-14}$$

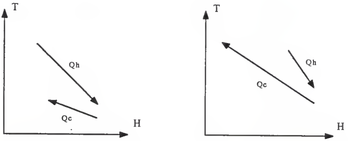
The elimination condition is

$$\Delta T > \Delta T_{min} \tag{4-15}$$

at all point in a match.

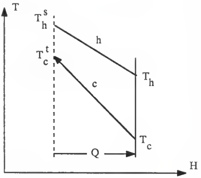


(a) hot end match (Qh > Qc) (b) hot end match (Qh < Qc)

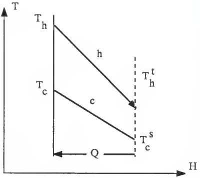


(c) cold end match (Qh > Qc) (d) cold end match (Qh < Qc)

Fig. 4-3. Possible ways to match a pair of streams.



(a) hot end match



(b) cold end match

Fig. 4-4. Driving force at an intermediate point in a match.

Hot End Match

The temperatures of the streams at an intermediate point, for a hot end match are given by

$$T_h = T_h^s - \frac{Q}{(mc_p)_h} \tag{4-16}$$

and

$$T_c = T_c^t - \frac{Q}{(mc_p)_c} \tag{4-17}$$

Hence, the driving force becomes

$$\Delta T = \left\{T_h^s - \frac{Q}{(mc_p)_h}\right\} - \left\{T_c^t - \frac{Q}{(mc_p)_c}\right\}$$

$$= \left[T_h^s - T_c^t\right] - Q\left[\frac{1}{(mc_p)_h} - \frac{1}{(mc_p)_c}\right] \tag{4-18}$$

From Eq. 4-12 it can be seen that the first term in the right hand side of Eq. 4-18 is nothing but the driving force at the hot end (ΔT_{he}). For elimination, the driving force given by Eq. 4-18 must not be less than the minimum allowable value for all the points in the match, i.e., from the beginning of the match till the end when the heat duty of one of the streams get exhausted. At the outset of the match,

$$Q = 0 \tag{4-19}$$

and at the end,

$$Q = \min(Q_h, Q_c) \tag{4-20}$$

Consequently, the condition for elimination, Eq. 4-15, on substitution of Eqs. 4-18 and 4-19, becomes

$$\Delta T_{he} \geqslant \Delta T_{min} \tag{4-21}$$

at the outset of the match, and on substitution of Eqs. 4-18 and 4-20, becomes

$$\left\{\Delta T_{he} - \min(Q_h, Q_c)\left[\frac{1}{(mc_p)_h} - \frac{1}{(mc_p)_c}\right]\right\} \geq \Delta T_{min} \tag{4-22}$$

at the termination of the match. In the limiting case, when ΔT_{he} equals ΔT_{min}, the condition in Eq. 4-22 reduces to

$$(mc_p)_h \geq (mc_p)_c \tag{4-23}$$

Cold End Match

The temperatures of the streams at an intermediate point, for a cold end match, are given by

$$T_h = T_h^t + \frac{Q}{(mc_p)_h} \tag{4-24}$$

and

$$T_c = T_c^s + \frac{Q}{(mc_p)_c} \tag{4-25}$$

Hence, the driving force becomes

$$\begin{aligned}\Delta T &= \left\{T_h^t + \frac{Q}{(mc_p)_h}\right\} - \left\{T_c^s + \frac{Q}{(mc_p)_c}\right\} \\ &= \left[T_h^t - T_c^s\right] + Q\left[\frac{1}{(mc_p)_h} - \frac{1}{(mc_p)_c}\right]\end{aligned} \tag{4-26}$$

Eq. 4-13 indicates that the first term is nothing but the driving force at the cold end, ΔT_{ce}. Once again, for elimination, the driving force given by Eq. 4-26 must not be less than the minimum allowable value for all the points in the match. Based on the same end values for Q given by Eqs. 4-19 and 4-20, the condition for elimination, Eq. 4-15, becomes

$$\Delta T_{ce} > \Delta T_{min} \tag{4-27}$$

at the outset of the match, and

$$\left\{\Delta T_{ce} + \min(Q_h, Q_c)\left[\frac{1}{(mc_p)_h} - \frac{1}{(mc_p)_c}\right]\right\} > \Delta T_{min} \tag{4-28}$$

at the end of the match. In the limiting case, when ΔT_{ce} equals ΔT_{min}, the condition in Eq. 4-28 reduces to

$$(mc_p)_c > (mc_p)_h \tag{4-29}$$

As stated earlier, the conditions given by Eqs. 4-21 and 6-25 are necessary, but not sufficient. The elimination conditions, given by Eqs. 4-21 through 4-23 and Eqs. 4-27 through 4-29, are nothing but the formalization and generalization of the matching criteria proposed by Linhoff et al. (1982). These criteria, identical to those in Eqs. 4-23 and 4-29, have been proposed for selection of the stream/stream matches on either side of a pinch point. Although the criteria are exactly the same, the approaches for arriving at them are entirely different; whereas Linhoff et al. (1982) have proposed theirs specifically and only for selecting matches at the pinch, the same conditions derived in this work are applicable for selecting matches whenever the approach temperature at any end of a match (ΔT_{he} or ΔT_{ce}) is equal to the minimum value (ΔT_{min}). The pinch point is only a specific instance of this equality; there may be situations other than the pinch point where the equality requirement is satisfied.

In addition, the Eqs. 4-23 and 4-29 can prove to be important short-cut conditions for developing a synthesis strategy; when the approach temperature (ΔT_{he} or ΔT_{ce}) are greater than the minimum value (ΔT_{min}), then these conditions are sufficient (but not necessary) to guarantee the elimination of one of the streams, thus allowing us to skip the computation of the left hand sides of the Eqs. 4-22 and 4-28. As we shall see in the subsequent sections, this observation is indeed very useful in devising a strategy for HEN synthesis.

The insight into the problem of minimum number of HTU's obtained in this section is entirely due to the rigorous formalization of the underlying knowledge. The failure of the previous workers, e.g., Linhoff _et_ _al_. (1982), can be attributed, in part, to their not "asking" the "correct" questions and to their reliance upon only the qualitative analysis.

CHAPTER 5. STREAM SPLITTING FOR HEAT EXCHANGER NETWORKS

Several results derived in Chapter 4 are based on the assumption that matches of type (c) can be entirely avoided during the synthesis of a HEN. In this chapter, we shall see how this can be achieved.

The necessity for creating type (c) matches arise when

(i) a certain stream can not "participate" in generating a type (a) or type (b) match with any other stream, and

(ii) a certain stream can "participate" in generating a type (a) or type (b) match, but doing so would lead to utility consumption in excess of the predicted target.

While we can proceed by generating a match of type (c), it will prevent the attainment of the minimum number of HTU's, as discussed in the preceeding chapter. However, by resorting to stream splitting, generation of type (c) matches can be prevented. By splitting a stream, we "replace" it by two or more substreams having lower values of heat capacity flow rates (mc_P values), thereby enhancing the possibility of satisfying the elimination criteria (developed in Chapter 4) for individual substreams. Thus, stream splitting enables us to continue HEN synthesis while adhering to the elimination strategy.

NEED FOR STREAM SPLITTING

Consider the problem depicted in Figure 5-1; this is the well-known 4SP2 problem (see, e.g., Ponton and Donaldson, 1974; Pehler and Liu, 1984) with all values of the parameters (temperatures and heat capacity flow rates) expressed in SI units, after rounding off. The problem has a pinch point at the left edge, at -4°C. Since the problem contains only the hot subnetwork (above the pinch), it requires only hot utility, with a minimum consumption target of 399.5 kW. The minimum driving force is specified to be 10°C.

We begin our synthesis by matching the hot utility. Obviously, the hot utility must be used for heating a cold stream. Adopting the stream/stream heat selection criteria HS1h/HS2c, (cf. Chapter 3) the hottest part of the hot stream (hot utility) will be matched with the hottest part of the cold stream. Since the problem involves only one cold stream, C1, a heater with a duty of 399.5 kW is "placed" on C1, as illustrated in Figure 5-2.

Applying the feasibility criteria for eliminating at least one stream (the elimination conditions, developed in the preceeding chapter), it can be readily seen that each of the two hot streams H1 and H2 can be matched with C1 at both ends (hot and cold), whereas the third hot stream, H3, can be matched with C1 only at the cold end; all matches are of type (b). Also, since no cold utility is required in the problem, each of the three hot streams must be cooled by matching with C1. Thus, a question arises; in what order should the streams be matched?

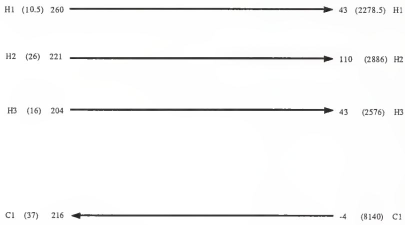


Fig. 5-1. 4SP2 problem (Ponton and Donaldson, 1974).

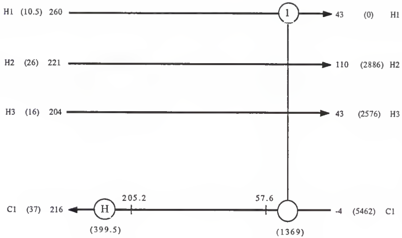


Fig. 5-2. A dead-end situation for the 4SP2 problem.

When C1 is matched with any of the three hot streams, H1, H2 and H3, its source temperature will increase from the "initial" value of -4°C. Obviously, the increment is directly proportional to the heat load of the HTU generated by the match. Consequently, the feasibility of subsequent matches may be affected since the driving force at the cold end is reduced. Let us minimize this affect by selecting the lowest heat load for this HTU, i.e., let us chose H1 which has the lowest value of Q among the three hot streams. The resultant match, as indicated in Figure 5-2, raises the source temperature of C1 from -4°C to 57.6°C.

Now it is not possible to match H3 with C1 such that H3 gets cooled to 43°C, since the driving force at the cold end is negative. This is reflected in the fact that the elimination condition (Eq. 4-27) is not satisfied for H3/C1 match. Thus we have reached a dead end; no type of match would enable us to get out of this situation (the target temperature of H3 can never be attained). The only way out is to use additional amount of cold utility and the corresponding amount of additional hot utility (to maintain the heat balance). Since we have chosen hot stream H3 with the lowest value of Q, any other stream, H2 or H1, will create a similar dead end situation, because the source temperature of C1 for these cases will be higher then 57.6°C, there by leaving one or both of streams, H1 and H3, "unmatchable".

One way to circumvent this difficulty, without consuming "extra" amounts of utilities, is to allow the H1/C1 match to be of type (c), i.e., match the streams only to the point when the source temperature of C1 is raised to a value just enough to cool H3, i.e., a value of 33°C.

Now H3 can be matched with C1. Again, if a type (b) match is generated, which is feasible at the cold end of the two streams, the "new" source temperature of C1 becomes 102.6°C; this is not sufficiently low (100°C or lower) to cool H2, as illustrated in Figure 5-3. Thus, once again we are at a dead end; no match of any type can enable H2 to attain its target temperature without consuming "extra" utilities, both, hot and cold. To get out of this predicament, we need to resort to the same approach that was employed earlier; the H3/C1 match should be of such magnitude that the "new" source temperature of C1 is 100°C.

Continuing in this fashion, we arrive at a solution shown in Figure 5-4; which has eight units. This network has four type (c) matches (matches 1, 2, 3, and 5) resulting in four extra HTU's than the minimum value of 4, predicted by Eq. 4-5. Several alternate HEN configurations are possible, but it is impossible to synthesize a network featuring simultaneously the minimum utility consumption and the minimum number of HTU's, without resorting to stream splitting. In other words, in situations like this one, stream splitting enables us to synthesize a HEN featuring both the minimum HTU's and the minimum utility consumption.

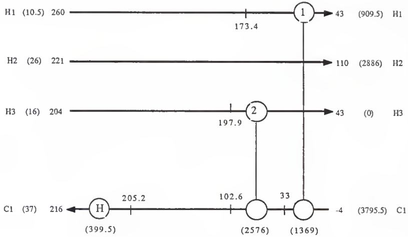


Fig. 5-3. Another dead-end situation for the 4SP2 problem.

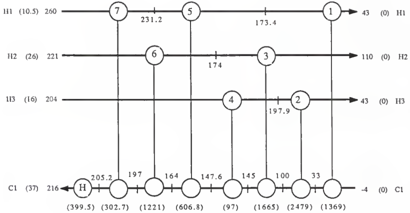


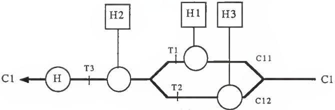
Fig. 5-4. A possible solution for the 4SP2 problem; featuring the minimum utility consumption without stream splitting.

AN EXERCISE IN STREAM SPLITTING

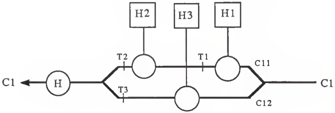
Having demonstrated the need for stream splitting, let us see how this task can be performed. A rather limited number of the synthesis strategies proposed thus far permit stream splitting. Further, those permitting the stream splitting lack the definite guidelines or rules to perform this task. Linhoff et al. (1982) have proposed algorithms for splitting streams at a pinch point; however, these algorithms are not concerned with stream splitting anywhere else in the network. Consequently, in a situation such as the one we are confronted with in solving the 4SP2 problem, their algorithms are useless.

Traditionally, the task of stream splitting has been left to the design engineer's *insight*, *experience*, and *creativity*; this task has been thought to be beyond the capabilities of computers. It is the ultimate goal of this work to demonstrate that, with a proper approach, it is possible to "teach" the machines (computers) how to perform such a task. Towards this end, it is essential that we attempt to formalize the *insight* and *experience* of expert designers to create a "knowledge bank" which can be accessed by humans or machines. This will not only facilitate the automation of HEN synthesis, but also enable the novice engineers/designers to gain access to the expertise of the expert designers. Let us explore the possibilities of stream splitting for the 4SP2 problem and see what insight it offers towards identifying the issues affecting the task of stream splitting.

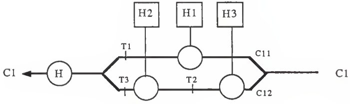
Figure 5-5 shows four possible splitting patterns for the 4SP2 problem. For brevity, the hot streams are not depicted; instead, the



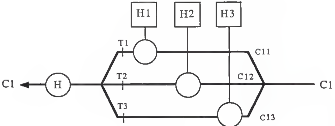
Pattern (a)



Pattren (b)



Pattern (c)



Pattern (d)

Fig. 5-5. Possible splitting patterns for the 4SP2 problem.

hot stream for each match is indicated as a label on the match. Also, the values of the heat loads and the temperatures are omitted since we are only concerned with the feasibility of the structural configuration of the split substreams.

Splitting is concerned with distributing the heat capacity flow rate (mc_p) of the original stream among the newly created substreams. For a pattern to be feasible, the elimination conditions derived in Chapter 4 must be satisfied for all matches involved in the stream splitting. These elimination conditions, Eqs. 4-21 through 4-23 and Eqs. 4-27 through 4-29, will define constraints on the mc_p values for the substreams. Specifically on rearrangement of Eqs. 4-22 and 4-28, we obtain

$$\left[\frac{1}{(mc_p)_h} - \frac{1}{(mc_p)_h}\right] < \frac{[\Delta T_{he} - \Delta T_{min}]}{Q} \tag{5-1}$$

for a hot end match, and

$$\left[\frac{1}{(mc_p)_c} - \frac{1}{(mc_p)_h}\right] < \frac{[\Delta T_{ce} - \Delta T_{min}]}{Q} \tag{5-2}$$

for a cold end match, respectively, where Q is the heat load of the HTU resulting from the match. If all the constraints are satisfied simultaneously, then the pattern is feasible; otherwise, it is not. Let us examine each of the four patterns shown in Figure 5-5 and establish its feasibility.

Consider pattern (a). The cold stream C1 is split into two substreams, $C1_1$ and $C1_2$, which are matched with H1 and H3, respectively. These streams are then merged back to form C1 and matched with H2. Note that the match with H2 "follows" the matches with H1 and H3 since the target temperature of H2, T^t_{H2}, is the highest and thus it can be attained by matching with a cold stream having a higher source temperature.

Applying the elimination conditions to H2/C1 match from the cold end, Eq. 4-27 gives

$$\Delta T_{ce} = T^t_{H2} - T_1 \geq 10$$

For $T^t_{H2} = 110°C$, this reduces to

$$T_1 \leq 100 \tag{5-3}$$

The intermediate temperature T_1 of stream C1 is given by the following heat balance equation;

$$(mc_P)_{C1}(T_1 - T^s_{C1}) = Q_{H1} + Q_{H3}$$

or,

$$T_1 = \frac{Q_{H1} + Q_{H3}}{(mc_P)_{C1}} + T^s_{C1}$$

Substituting the values into the right hand side of this equation, we obtain

$$T_1 = \frac{2278.5 + 2576}{37} + (-4)$$

$$= 127.2°C \tag{5-4}$$

Since T_1 exceeds the maximum allowable value of 100°C, we conclude that pattern (a) is infeasible.

Next, consider pattern (b). Again, the cold stream C1 is split into two substreams, $C1_1$ and $C1_2$. This time, however, $C1_1$ is matched with H1, "followed" by the match with H2, whereas $C1_2$ is matched with H3, as in pattern (a). Pattern (b) is feasible if and only if the elimination conditions are satisfied for all the matches.

For the $H2/C1_1$ match, the elimination conditions, based on the cold end matching are

$$\Delta T_{ce} = T^t_{H2} - T_2 > 10$$

or, on substitution

$$T_2 < 100 \tag{5-5}$$

at the outset of the match, and

$$\left[\frac{1}{(mc_P)_{C1_1}} - \frac{1}{(mc_P)_{H2}}\right] < \frac{\Delta T_{ce} - \Delta T_{min}}{Q_{H2}}$$

or, on substitution

$$\frac{1}{(mc_P)_{C1_1}} < \frac{(110 - T_2) - 10}{2886} + \frac{1}{26} \tag{5-6}$$

at the termination of the match. The intermediate temperature T_2 is given by the following heat balance relationship;

$$(mc_P)_{C1_1} (T_2 - T^s_{C1}) = Q_{H1}$$

which, on substitution and simplification, yields

$$T_2 = \frac{2278.5}{(mc_P)_{C1_1}} - 4 \tag{5-7}$$

From Eqs. 5-5 and 5-7, we obtain

$$(mc_P)_{C1_1} > 19.99 \tag{5-8}$$

Eliminating T_2 from Eqs. 5-6 and 5-7, and solving for $(mc_P)_{C1_1}$, we have

$$(mc_P)_{C1_1} > 24.02 \tag{5-9}$$

For the H1/C1_1 match, the elimination condition at the beginning of the match is satisfied since

$$\Delta T_{ce} = T^t_{H1} - T^s_{C1} = 43 - (-4) = 47$$

which is indeed greater than the minimum value of 10°C. At the end of the match, the elimination condition is given by Eq. 5-2, for this match, it reduces to

$$\left[\frac{1}{(mc_P)_{C1_1}} - \frac{1}{(mc_P)_{H2}}\right] < \frac{\left(T^t_{H1} - T^s_{C1}\right) - \Delta T_{min}}{Q_{H1}}$$

which on substitution and simplification, yields

$$(mc_P)_{C1_1} > 8.97 \tag{5-10}$$

For the H3/C1_2 match, the elimination conditions are

$$\Delta T_{ce} = T^t_{H3} - T^s_{C1} = 43 - (-4) = 47 > 10$$

which is true, and

$$\left[\frac{1}{(mc_P)_{C1_2}} - \frac{1}{(mc_P)_{H3}}\right] < \frac{\left(T^t_{H3} - T^s_{C1}\right) - \Delta T_{min}}{Q_{H3}}$$

which, on substitution and simplification, yields

$$(mc_P)_{C1_2} > 13.01 \tag{5-11}$$

For this pattern to be feasible, the constraints on mc_p values, obtained in Eqs. 5-8 through 5-11, must be satisfied. Eqs. 5-8 through 5-10 can be combined to form the following single feasible region for $(mc_p)_{C1_1}$;

$$(mc_p)_{C1_1} \geq 24.02 \tag{5-12}$$

Since $C1_1$ and $C1_2$ are substreams of C1, we have

$$(mc_p)_{C1_1} + (mc_p)_{C1_2} = 37 \tag{5-13}$$

From Eqs. 5-11 through 5-13, we can conclude that pattern (b) is infeasible. However, if we round off the values, we see the existence of a singular point corresponding to the values

$$(mc_p)_{C1_1} = 24$$

and

$$(mc_p)_{C1_2} = 13$$

The resultant network is shown in Figure 5-6. Note that the driving force at the hot end of the $H2/C1_1$ match is 9.81°C, just under the minimum allowable value of 10°C. Thus, even though the solution violates slightly the constraint of ΔT_{min} it is acceptable.

Pattern (c) is similar to pattern (b) except that H2 is now matched with $C1_2$, following the $H3/C1_2$ match. Analysis, similar to that for the pattern (b), yields the following results.

For the $H2/C1_2$ match the elimination conditions at the outset and at the termination result in

$$(mc_p)_{C1_2} \geq 24.7 \tag{5-14}$$

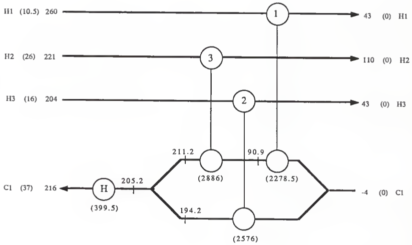


Fig. 5-6. Solution for the 4SP2 problem with pattern (b).

and

$$(mc_p)_{C1_2} \geq 19.2 \tag{5-15}$$

respectively. For the H3/C1$_2$ match, the elimination condition at the beginning of the match is satisfied (since ΔT_{ce} = 47°C > 10°C), and that at the end of the match results in the following feasible region;

$$(mc_p)_{C1_2} \geq 13.01 \tag{5-16}$$

For the H1/C1$_1$ match, once again the condition at the beginning of the match is satisfied (since ΔT_{ce} = 47°C > 10°C) and the condition at the end of the match reduces to

$$(mc_p)_{C1_1} \geq 8.97 \tag{5-17}$$

Once again, for the substreams C1$_1$ and C1$_2$, we have

$$(mc_p)_{C1_1} + (mc_p)_{C1_2} = 37 \tag{5-18}$$

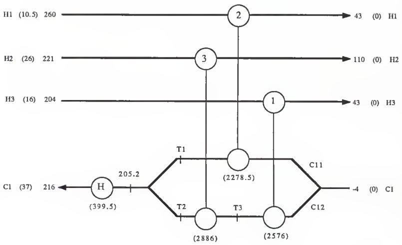
Based on Eqs. 5-14 through 5-18, we can conclude that the feasible regions for the mc_p values are

$$8.97 \leq (mc_p)_{C1_1} \leq 12.3 \tag{5-19}$$

and

$$24.7 \leq (mc_p)_{C1_2} \leq 28.03 \tag{5-20}$$

subject to the constraint given by Eq. 5-18. The resultant network is shown in Figure 5-7 with four possible values of $(mc_p)_{C1_2}$: 25, 28, 27, and 28.



Heat capacity flow rates		T1	T2	T3
C11	C12			
12.0	25.0	185.9	214.5	99.0
11.0	26.0	203.1	206.1	95.1
10.0	27.0	223.9	198.3	91.4
9.0	28.0	243.2	191.1	88.0

Fig. 5-7. Solution for the 4SP2 problem with pattern (c).

Finally, pattern (d) splits up the stream C1 into three substreams, $C1_1$, $C1_2$, and $C1_3$, matched with H1, H2, and H3, respectively. Once again, carrying out the same analysis as in the previous case, we see that the elimination conditions are satisfied at the beginning of all three matches (ΔT_{ce} = 47°C). Using Eq. 5-2 for the elimination conditions at the end of all the matches, we obtain the following constraints. For the match $H1/C1_1$,

$$(mc_P)_{C1_1} \geq 8.97, \tag{5-21}$$

for the match $H2/C1_2$

$$(mc_P)_{C1_2} \geq 13.42, \tag{5-22}$$

and for the match $H3/C1_3$,

$$(mc_P)_{C1_3} \geq 13.01 \tag{5-23}$$

On summation of Eqs. 5-21 through 5-23, we obtain

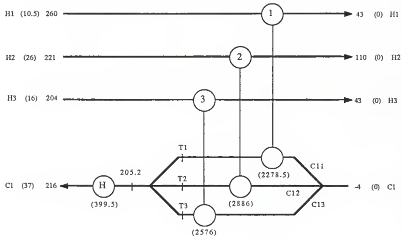
$$(mc_P)_{C1_1} + (mc_P)_{C1_2} + (mc_P)_{C1_3} \geq 35.4 \tag{5-24}$$

Also, for this pattern, we know that

$$(mc_P)_{C1_1} = (mc_P)_{C1_2} + (mc_P)_{C1_3} = 37 \tag{5-25}$$

Based on these last two results, we conclude that pattern (d) is feasible, subject to the constraints given by Eqs. 5-21 through 5-25. The resultant network is depicted in Figure 5-8, with mc_P value triplets (9, 13.5, 14.5), (9, 14, 14), (9, 14.5, 13.5), and (10, 13.5, 13.5).

One of the simplest ways of distributing the heat capacity flow rates among the substreams is to make the temperature drop for all the



Heat capacity flow rates			T1	T2	T3
C11	C12	C13			
9.0	13.5	14.5	249.2	209.8	173.7
9.0	14.0	14.0	249.2	202.1	180.0
9.0	14.0	13.5	249.2	195.0	186.8
10.0	13.5	13.5	223.9	209.8	186.8

Fig. 5-8. Solution for the 4SP2 problem with pattern (d).

substreams to be equal. Such a situation can be arrived at by distributing $(mc_p)_{C1}$ in the ratios of the heat load on each branch (substream). This condition can be expressed as follows

$$\frac{(mc_p)_{c_i}}{(mc_p)_c} = \frac{Q_{c_i}}{Q_c} \tag{5-26}$$

where c_i is the i th substream of the cold stream c, Q_{c_i} is the summation of the heat loads of all the HTU's on c_i and Q_c is the summation of the heat loads of all the HTU's on all the substreams of c. Applying this criterion to the three feasible patterns for the 4SP2 problem, we obtain the following distributions:

For the pattern (b),

$(mc_p)_{C1_1} = 24.69$ and $(mc_p)_{C1_2} = 12.31$,

for pattern (c),

$(mc_p)_{C1_1} = 10.89$ and $(mc_p)_{C1_2} = 26.11$,

and finally, for pattern (c),

$(mc_p)_{C1_1} = 10.89$, $(mc_p)_{C1_2} = 13.80$, and $(mc_p)_{C1_3} = 12.31$

Comparison with the feasible regions for the mc_p values established earlier in this section, it can be seen that of the above distributions, only the one corresponding to pattern (b) is feasible.

SYSTEMATIC PROCEDURE FOR STREAM SPLITTING

The insights obtained by solving the 4SP2 problem can be utilized to formulate a systematic approach to stream splitting. Before proceeding any further in this direction, let us define some terms which will facilitate the description of the approach.

All the streams (hot and cold) that take part in splitting will be termed as *participating streams*. These streams are further classified into two categories; the streams to be split (usually one per splitting), termed as the *candidate streams*, and the streams that "force" the splitting, termed as the *competing streams*. The substreams resulting from the splitting of a candidate stream shall be termed as the *candidate substreams*. Note that the candidate stream (as well as the candidate substreams) and the competing streams will always be of the opposite kinds, hot or cold.

A *splitting (or split) pattern* refers to an arrangement of matches between the competing streams and the candidate substreams; the pattern may or may not be feasible. Several patterns can be generated for a given set of participating streams; depending upon the source/target temperatures and the heat capacity flow rates of the streams involved, any number of these patterns may be feasible.

A particular pattern of interest is the one which contains as many candidate substreams as the number of competing streams. Each candidate substream has exactly one match. In other words, each match "eliminates" one competing stream and one candidate stream. This pattern will be termed as the *fully split pattern*. Note that for a given set of participating streams there exists a unique fully split pattern. Any pattern that has fewer candidate substreams than the number of competing streams, will be termed as a *partially split pattern*. Obviously, a partially split pattern will have at least one candidate substream which has multiple matches. The process of transforming a fully split pattern into a partially split pattern will be termed as *folding*, and the reverse process, *unfolding*. The subnetwork formed by matching the participating streams according to a pattern will be called *split subnetwork*. Each feasible split pattern will give rise to a split subnetwork; the subnetworks obtained from the fully and partially split patterns will be called the *fully split subnetwork* and *partially split subnetwork*, respectively. Armed with this set of terminologies, we are ready to develop a systematic approach to stream splitting.

First we shall identify the situations where stream splitting is required. On examining our test case, the 4SP2 problem from the preceeding section, it can be seen that the problem required us to match more than one hot stream with one cold stream. However, we could not generate matches of type (a) or (b). More specifically, creating one match left the remaining streams without any match, leading to a dead-end situation. This observation enables us to propose the following guideline for identifying the situation warranting stream splitting.

When more than one streams of any kind (hot or cold) are required to be matched with only one stream of the opposite kind (cold or hot) and their source/target temperatures are such that sequential matches are *not feasible while adhering to the elimination strategy, then we need to split the latter, i.e., the stream of the opposite kind.*

This will not only indicate when to split a stream, but also enable us to identify the candidate stream (the one to be split) and the competing streams (the streams that "force" the splitting).

Second, we need to know how to split a candidate stream, once it has been identified. Four splitting patterns were obtained for our test case, the 4SP2 problem in the preceeding section. As the complexity of the problem increases, we may have several options. How do we choose a pattern? Again, based on our experience with the 4SP2 problem, we can propose that we should choose a splitting pattern that allows us to continue HEN synthesis by adhering to the elimination strategy. Obviously, the pattern should be a feasible one.

HEN synthesis can be continued with the elimination strategy if we ensure that each match in the split subnetwork eliminates a competing stream. All four patterns considered for the 4SP2 problem fall into this category. As expected, all three feasible patterns lead to HEN's featuring the minimum number of units, i.e., 4 (see Figures 5-6, 5-7, and 5-8).

Feasibility of a splitting pattern is established by examining whether or not the elimination conditions (Eqs. 4-21 and 4-22 for the hot end matching and Eqs. 4-27 and 4-28 for the cold end matching) are satisfied for all the matches in the pattern. These conditions can be satisfied unconditionally (evaluate to be true for any mc_p value) or conditionally (evaluate to be true only for a specific range of mc_p values). In other words, the elimination conditions establish a range of mc_p values, for which the pattern is feasible; this range is null when the condition is not satisfied. If there exist a set of mc_p values (one each for for all the candidate substreams), all of which fall into the respective allowable ranges, and if they satisfy the constraint that sum of the mc_p values for all the substreams must equal to the mc_p value of the candidate stream, then the corresponding pattern is feasible.

Thus for any feasible pattern, there exists a range of mc_p values that the candidate substreams can have. How should we choose a set of values in order to generate a split subnetwork? In the absence of any external constraint, we can adopt an *equal temperature difference* policy. This policy mandates that the mc_p values of the candidate substreams should be chosen such that the temperature differences across all the candidate substreams be equal. This ensures that when the substreams are mixed (combined) to form the original stream, the mixing will be isothermal. Mixing streams of varying temperatures is thermodynamically inefficient, since it results in the dissipation of available energy. According to the equal temperature difference policy, the heat capacity flow rate of j th candidate substream, $(mc_p)_{ij}$, is given by

$$(mc_p)_{i_j} = \frac{Q_{i_j}}{Q_i} (mc_p)_i \qquad (i = c,\ h;\ j = 1,\ 2,\ \dots\ \text{etc.}) \qquad (5\text{-}27)$$

where i represents the the stream to be split (the candidate stream), cold or hot, and $(mc_p)_i$ is the heat capacity flow rate of the candidate stream. Q_{i_j} is the total heat load on the j th substream; it is simply the sum of the heat duties of all the HTU's involving the j th substream, that will be generated by the pattern. Q_i is the summation of heat loads on all the candidate substreams, and is given by

$$Q_i = \sum_{j=1}^{m} Q_{i_j} \tag{5-28}$$

In the event that the values obtained by using Eq. 5-27 do not fall into the feasible regions, the values from the feasible regions closest to the ones obtained using the equal temperature difference policy (i.e., using Eq. 5-27 can be selected.

Having established the criterion that a feasible pattern is needed to continue the HEN synthesis based on the elimination strategy, we shall still be left with several patterns to choose from. The choice of pattern can have profound effects on some of the structural properties of the resultant HEN, including the total heat transfer area, resiliency, operability, and controllability. Since for any given set of participating streams, there exist a unique fully split pattern, it seems to be a reasonable choice to start with; we do not have to worry about how to generate various patterns. We can simply split the candidate stream into as many substreams as the number of competing streams and match each substream with one of the competing streams.

Furthermore, a fully split pattern , if feasible, will always result in a HEN with better properties than those of a HEN resulting from a partially split pattern; in comparision with the latter, the former possesses lesser heat transfer area because of higher driving force, greater resilience because of the reduced effect of load fluctuations which are "localized" to only one match within the split subnetwork, better operability and controllability because all matches in the split subnetwork are "independent" of each other. By "independence", we mean that the performance of a match on one substream does not affect the performance of the matches on the rest of the substreams. Therefore, we shall always try and obtain a fully split solution.

If a fully split pattern is not feasible (which is quite possible), then we need to fold the pattern into a partially split pattern by "pooling" two matches on one substream, thus reducing the number of candidate substreams by 1. If the pattern is still not feasible, then further folding is required. Folding can be performed in many ways; it is a combinatorial problem, with possible combinations depending upon the source/target temperatures and the mc_p values of the participating streams. Systematic guidelines for folding (a fully split pattern) can only be obtained by analyzing the consequences of folding on the structural properties of the resultant HEN. However, this task is too complex to be included in the scope of this work. As far as this work is concerned, if a fully split pattern is not feasible, then folding will be performed in an ad hoc fashion based on the source/target temperatures and the heat capacity flow rates of the streams involved. No attempt will be made to obtain an "optimally" folded pattern, or the "best" partially split pattern.

The proposed procedure for stream splitting can be summarized as follows:

Step 1. Identify the streams involved in the splitting, including the candidate stream and the competing streams.

Step 2. Generate a fully split pattern and examine its feasibility by employing the elimination conditions.

Step 3. If the pattern is feasible, determine the mc_p values for the substreams using the equal temperature difference policy, create the corresponding split subnetwork, and continue with the synthesis of the remainder of the network; otherwise proceed to step 4.

Step 4. Create a feasible partially split pattern and the corresponding split subnetwork by folding the fully split pattern.

As mentioned previously, the folding in step 4 will be performed in an ad hoc fashion, based on the heat capacity flow rates, heat loads and source/target temperatures of the participating streams. As additional knowledge is gained, this ad hoc approach to folding will be replaced by a systematic procedure to identify an optimally folded pattern.

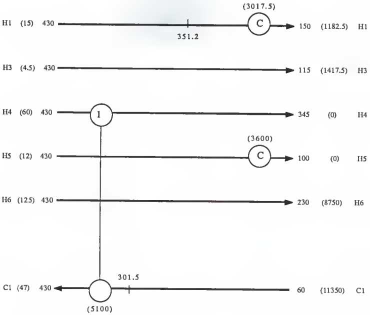
CHAPTER 6. REPRESENTATION AND MANIPULATION OF KNOWLEDGE FOR HEN SYNTHESIS

Having extracted and formalized some of the knowledge pertaining to the two aspects of HEN synthesis, viz., the minimum number of units and stream splitting, we are in a position to explore how this knowledge can be utilized effectively. To accomplish this, we require schemes to represent and manipulate this knowledge in an appropriate fashion. It is worth noting that we seek to represent symbols and associated concepts, not merely the numeric values. Further, the schemes should be capable of being utilized by both men and machines.

REPRESENTATION SCHEME

The proposed representation scheme, as illustrated in Figure 6-1, consists of two parts: a grid diagram and a match matrix. Figure 6-1(a) shows a typical grid diagram. The grid diagram representation and the information it portrays have been discussed in Chapter 4; hence it will not be repeated here.

Figure 6-1(b) shows the match matrix corresponding to the grid diagram in Figure 6-1(a). Each row of the match matrix contains the match information for a cold stream, and each column, the match information for a hot stream; the rows and the columns are labeled with the corresponding stream "names". In other words, each entry ("box") in the matrix displays the information regarding the match between the cold and hot stream corresponding to the row and column to which it belongs. If a match already exists between two streams, then the corresponding "box" contains the heat duty of the resultant HTU; otherwise, it indicates the feasibility of matching the streams. The feasibility of a hot end match is denoted by "H" and that of a cold end match, by "C". The infeasibility for both, hot and cold end matches, is denoted by an asterisk (star,"*"). The feasibility (or infeasibility) is established using the elimination conditions developed in Chapter 4. If the heat load of a stream is fully satisfied, then it no longer needs to be considered for subsequent matches. For such a stream, the match matrix contains a dash ("—") in each "box" in the corresponding column (for a hot stream) or row (for a cold stream), except for the "boxes" containing the heat load(s) of HTU's.



(a) grid diagram

Hot / Cold	H1	H3	H4	H5	H6	Qc
C1	H C	* C	5100	---	* C	11350
CU	3017.5	---	---	3600	---	0
Qh	1182.5	1417.5	0	0	8750	11350

(b) match matrix

Fig. 6-1. Representation scheme for HEN synthesis.

The last row, labeled Q_h, contains the unsatisfied or residual heat loads of the hot streams (i.e., the heat loads yet to be satisfied). Similarly, the last column, labeled Q_c, contains the unsatisfied or residual heat loads for the cold streams. Additionally, the last element in the match matrix, corresponding to row Q_h and column Q_c, contains the total amount of heat integration yet to be carried out in the problem. It is simply the sum of all the values in the corresponding row or column.

The match matrix in Figure 6-1(b) presents the following information on the "current" status of the problem.

(i) The partial solution (network) consists of three HTU's; one cooler each on streams H1 and H5, with heat duties of 3017.5 and 3600 units respectively, and a heat exchanger for H4/C1 match, with a heat duty of 5100 units.

(ii) Three streams have been "eliminated" from the problem: cold utility Cu and hot streams H4 and H5.

(iii) H1/C1 match is feasible at both, hot and cold, ends whereas H3/C1 and H6/C1 matches are feasible only at the cold ends.

(i) and (ii) indicate that so far, the solution has been able to adhere to the elimination strategy. In other words, upto this point in synthesis, the solution has managed to have only the minimum number of HTU's, independent of the strategy employed in arriving at this solution. The information contained in (iii) facilitates in deciding which streams should be matched next. Thus, from this example, it is clear that a match matrix displays the current status of a problem (i.e., the partial solution attained until a given instant); it plays an important role in charting the path for the rest of the HEN synthesis task.

As part of the entire representation scheme, the grid diagram displays the spatial configuration of the HEN, the chronological order of the matches (i.e., the order in which the matches have been selected), and the intermediate temperatures of the streams between the matches. All this information can not be obtained from the match matrix. Note that it is possible to construct a match matrix from a grid diagram, but not vice-versa. To construct a grid diagram from a match matrix, we need to know the order in which the matches have been generated and at what end each match is made (hot or cold end). Together, the match matrix and the grid diagram provide us with a representation scheme that is fairly complete and powerful. As discussed in the next sections, the scheme provides us with a vehicle to experiment for the purpose of extracting and formalizing additional knowledge.

MANIPULATION SCHEME

Equipped with the knowledge and the representation scheme, we are in a position to proceed to synthesize a HEN. The knowledge extracted and formalized thus far deals with ways to attain the minimum utility target with the minimum number of HTU's. The elimination conditions derived in Chapter 4 specify the matches that will violate these constraints; they do not say anything about which match should be chosen out of all the feasible ones. To develop a knowledge-based system for HEN synthesis, it is imperative that a selection strategy be developed. However, to develop such a strategy, a considerable amount of experimentation is required; several candidate strategies need be evaluated by solving as many HEN problems as possible. Towards this end, a scheme to manipulate the existing knowledge is required: a scheme that enables us to utilize the knowledge formalized in the present work, and simultaneously, capable of accomodating additional knowledge. A manipulation scheme that meets these requirements is proposed in this section. The synthesis of a HEN can be carried out in two stages: preanalysis and network invention. The proposed manipulation scheme is to be employed in the second state, i.e., for network invention.

Preanalysis deals with identifying the location of the pinch point and determining the minimum utility requirement for the problem under consideration. It is not required to set the minimum units target as long as the network invention strategy adheres to the elimination criterion. In the course of the present work, all the illustrative problems have been taken from the literature where they have been preanalyzed and solved. Consequently, for these problems, the reported results of preanalysis will be used. Problems for which these results are not available, preanalysis can be performed using the temperature-interval (TI) method (See, e.g., Linhoff and Flower, 1978a, Linhoff et al., 1982).

A select-match-update cycle is proposed as the knowledge manipulating scheme for network invention. First, based on the information contained in the match matrix, a pair of streams (a hot and a cold stream) are selected for matching. Next, these streams are matched to generate an HTU on the grid diagram. Finally, the match matrix is updated to reflect the changes in the feasibilities of various possible matches. Repeated application of this select-match-update cycle will generate a HEN featuring the minimum number of HTU's and the minimum utility requirement.

Selection of a pair of streams to be matched involves two types of knowledge: problem-specific and domain-specific. Both types of knowledge require considerable experiential expertise. The problem-specific knowledge has already been incorporated in the representation scheme, in the form of a match matrix. The domain-specific knowledge, dealing with ways of utilizing the problem-specific knowledge, exists in an empirical form; to formalize it is a task of paramount difficulty; it is beyond the scope of the present work. In lieu of any formal knowledge, this step can be performed in an ad hoc fashion by a designer. Once substantial expertise is gained by solving several problems, a systematic strategy can be formulated, which can then be used by machines.

After selecting the streams to be matched, the next step in the select-match-update cycle is to "make" the match on the grid diagram. Since the manipulation scheme adheres to the elimination criterion, the heat load of one of the two streams (the "smaller" one) gets fully satisfied, thereby eliminating it from further consideration. The temperature of the remaining stream (the "larger" one) will change at the end at which the match is made, i.e., for a hot end match, the hot end temperature will change and for a cold end match, the cold end temperature will change. This "new" temperature is calculated and displayed on the grid diagram, along with the newly created HTU (e.g., the H4/C1 match in Figure 6-1(a), which is a hot end match, changes the hot end temperature of C1 to 301.5°F).

The last step in the select-match-update cycle is to update the match matrix. The feasibilities of all the matches, involving either of the streams matched in the preceding step (the "match" part of the cycle), change as a result of the match. The "box" corresponding to the current match now contains the heat duty of the HTU generated as a result of the match [see, e.g., H4/C1 match in Figure 6-1(a)]. The remaining "boxes" corresponding to the "eliminated" stream now contain the dashes ("—"). Also, the values of heat loads for the matched streams (Q_c and Q_h) get reduced by an amount equal to the heat duty of the resultant HTU. Lastly, the feasibilities for the matches involving the larger of the two streams matched (the one that does not get eliminated) are reevaluated. Note that the feasibilities of only those matches, which involve the uneliminated stream, need be reevaluated, since the feasibility of a match at any end depends only upon the driving force at that end, the heat loads of the streams and the heat capacity flow rates of the streams. Thus, the feasibility of a match is independent of the rest of the problem and therefore, all the feasibilities in a match matrix need not be reevaluated for every select-match-update cycle. For example, in Figure 6-1(b), only the feasibilities of H1/C1, H3/C1 and H6/C1 have been reevaluated.

At times, the select-match-update cycle may end up in a dead-end situation where a particular stream can not be matched with any other stream without consuming any "extra" utility or without violating the elimination criterion. Such a situation can easily be detected from the match matrix, which contains at least one stream which has an unsatisfied heat load and no feasible matches, i.e., all the "boxes" corresponding to that stream contain two *'s. In such cases, we need to backtrack and "undo" the last match, select an alternate pair of streams for matching, and continue with the select-match-update cycle to synthesize the HEN. This backtracking is not restricted to one step; any number of steps can be retraced and matches undone, depending upon the need.

The ability to backtrack is necessary, but not sufficient to guarantee a solution that satisfies the constraints of the minimum utility and the minimum number of HTU's. There may arise a situation where all the alternate match selections lead to the dead-end situation. In such cases, stream splitting is required. The participating streams can be easily identified; they include all the streams that form the alternate matches leading to the dead-end situation. The stream splitting is carried out as described the preceeding chapter. After generating the split subnetwork, the synthesis is continued using the select-match-update cycle. Note that the stream splitting requires creation of additional rows or columns, depending upon whether a cold stream is split or a hot stream is. This is accomplished simply by dividing the corresponding "boxes" in the match matrix to enter the heat duties of the HTU's constituting the split subnetwork.

The ability to perform stream splitting, in conjunction with the backtracking facility guarantees that a HEN that features both, the minimum utility consumption and the minimum number of HTU's can always be synthesized using the proposed manipulation scheme. The overall procedure for HEN synthesis can be summarized as the following sequence of steps.

1. Determine the pinch point location and the minimum utility requirement using the temperature-interval (TI) method. Decompose the problem at the pinch point and synthesize the two subnetworks independently by following steps 2 through 6. Obviously, this decomposition is not required for the problems having the pinch point at one end of the problem.
2. Create the initial grid diagram and the match matrix for the problem.
3. Select a pair of streams based on the information contained in the match matrix. In absence of any formal selection strategy, this task can be carried out in an ad hoc fashion.
4. Make the selected match on the grid diagram, recomputing the values that change as a consequence of the match.

5. Update the match matrix by reevaluating the feasibilities of the matches involving the stream not eliminated in step 4.
6. If updating in step 5 results in a dead-end situation, then go to step 7; otherwise repeat steps 3 through 6 until the entire HEN is synthesized.
7. "Undo" the match created in step 4, restore the grid diagram and the match matrix, select an alternate pair of streams to match (other than those pairs that have already been found to lead to the dead-end situation). Return to step 4 and continue the synthesis. If no such alternate pair can be found then go to the next step, step 8.
8. Identify the participating streams and perform the stream splitting. After generating a split subnetwork, go to step 5 and continue the synthesis procedure.

AN EXERCISE IN HEN SYNTHESIS: THE 7SP4 PROBLEM

This section will show how the proposed representation and manipulation schemes can be employed to synthesize a HEN. Figure 6-2 shows the problem selected for this purpose, the 7SP4 problem. The problem has been taken from Papoulias and Grossmann (1982). The pinch point is located at 430°F for hot streams and 410°F for cold streams. The problem requires 8390 Btu of hot utility and 6617.5 Btu of cold utility. The problem is split into two parts at the pinch point and the two parts are synthesized independently; the two resultant subnetworks are integrated to generate the overall solution.

Part I. Synthesis of the Above Pinch (Hot) Subnetwork

Step 1. Figure 6-3(a) shows the initial grid diagram for the problem, which consists of three hot streams, H1, H2, and H3, one cold stream, C1, and the hot utility Hu. Figure 6-3(b) shows the corresponding match matrix.

Step 2. Since there is only one cold stream (C1), the hot utility must be matched with this cold stream. The match has to be at the hot end since the hot utility must be supplied at the highest temperature. Thus, the first match is between C1 and Hu, at the hot end, with the resultant HTU having a heat duty of 8390 Btu. The smaller of the two streams, Hu, gets eliminated and the target temperature of C1 changes to a value given by

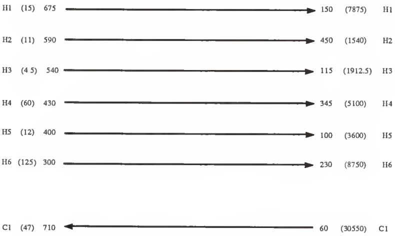
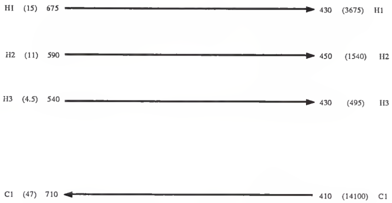


Fig. 6-2. 7SP4 problem: the initial grid diagram.



(a) grid diagram

Hot / Cold	H1	H2	H3	Hu	Qc
C1	* C	* C	* C	H *	14100
Qh	3675	1540	495	8390	14100

(b) match matrix

Fig. 6-3. Hot subnetwork for 7SP4 problem:
at the onset of synthesis (step 2).

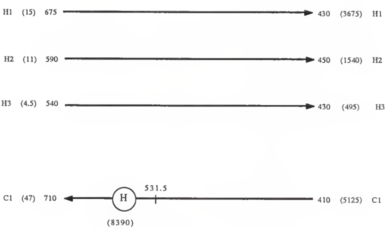
$$T^t_{C1} = 710 - \frac{8390}{47} = 531.5°F$$

The resultant problem status is shown in Figure 6-4(a) in the form of a grid diagram. The modified match matrix is shown in Figure 6-4(b).

Step 3. Next, all three hot streams need to be matched with C1. Since none of the hot streams, H1, H2, and H3, can be matched with C1 at the hot end, and C1 must have at least one match at the hot end in order to attain its target temperature, we can not make sequential matches. For illustration, making the H3/C1 match at the cold end would result in a situation depicted in Figure 6-5, with no matches possible for H1 at the hot end. Similarly, making H2/C1 or H1/C1 match also leads to a dead-end situation. This indicates that stream splitting is required.

Step 4. The stram splitting is carried out according to the procedure described in Chapter 5. The competing streams are, H1, H2, and H3, whereas C1 is the candidate stream. For the fully split pattern, C1 needs to be split into three substreams, $C1_1$, $C1_2$, and $C1_3$, to be matched with H1, H2, and H3, respectively. Applying the elimination conditions to these matches, we obtain the following feasible regions for mc_p values for the candidate substreams.

$$(mc_p)_{C1_1} \geq 15 \qquad (6\text{-}1)$$

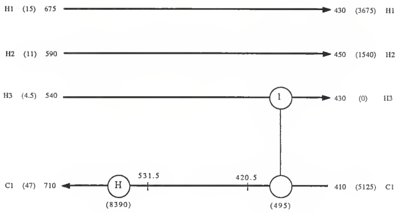


(a) grid diagram

Hot Cold	H1	H2	H3	Hu	Qc
C1	* C	* C	* C	8390	5710
Qh	3675	1540	495	0	5710

(b) match matrix

Fig. 6-4. Hot subnetwork for 7SP4 problem: after the Hu/C1 match (step 2).



(a) grid diagram

Hot Cold	H1	H2	H3	Hu	Qc
C1	* *	* C	495	8390	5215
Qh	3675	1540	0	0	5215

(b) match matrix

Fig. 6-5. Hot subnetwork for 7SP4 problem:
after the H3/C1 match (step 3).

$$(mc_P)_{C1_2} \geq 9.625 \qquad (6\text{-}2)$$

and

$$(mc_P)_{C1_3} \geq 4.5 \qquad (6\text{-}3)$$

Summation of Eqs. 6-1, 6-2, and 6-3, gives rise to the following feasibility condition;

$$(mc_P)_{C1_1} + (mc_P)_{C1_2} + (mc_P)_{C1_3} \geq 29.125 \qquad (6\text{-}4)$$

which holds since the left-hand side is $(mc_P)_{C1}$ which has a value 47, greater than 29.125. Thus, we conclude that a fully split pattern is feasible and folding is not required. For equal temperature drop across all three candidate substreams, the $(mc_P)_{C1}$ should be divided among the substreams in the ratio of the heat loads. Therefore, from Eq. 5-26, we obtain

$$(mc_P)_{C1_1} = \frac{Q_{H1}}{Q_{C1}} (mc_P)_{C1} = \frac{3675}{5710} (47) = 30.25 \qquad (6\text{-}5)$$

$$(mc_P)_{C1_2} = \frac{Q_{H2}}{Q_{C1}} (mc_P)_{C1} = \frac{1540}{5710} (47) = 12.68 \qquad (6\text{-}6)$$

and

$$(mc_P)_{C1_3} = \frac{Q_{H3}}{Q_{C1}} (mc_P)_{C1} = \frac{495}{5710} (47) = 4.07 \qquad (6\text{-}7)$$

The feasible regions of mc_P values given by Eqs. 6-1, 6-2, and 6-3, readily indicate that the values obtained in 6-5, 6-6, and 6-7, render the pattern infeasible. However, based on Eqs. 6-1 through 6-7, we can arrive at the following compromise values;

$$(mc_P)_{C1_1} = 30.0$$

$$(mc_P)_{C1_2} = 12.5$$

and

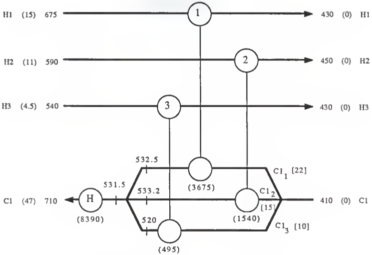
$$(mc_P)_{C1_3} = 4.5$$

Corresponding solution is shown in Figure 6-6. This completes the synthesis of the above-pinch (hot) subnetwork.

Part II. Synthesis of the Below Pinch (Cold) Subnetwork

Step 5. Figure 6-7(a) shows the initial grid diagram for the problem, consisting of five hot streams, H1, H3, H4, H5 and H6, one cold stream C1 and the cold utility Cu. The corresponding match matrix is shown in Figure 6-7(b).

Step 6. Out of the five hot streams, only H4 can be matched with C1 at the hot end. H5 and H6 do not have high enough source temperature ($\Delta T_{he} < \Delta T_{min}$ for matches with C1); whereas H1 and H3 do not have high enough (mc_P) values [for a hot end match at pinch point, $(mc_P)_h \geqslant (mc_P)_c$]. Hence, the H4/C1 match is made, thereby eliminating H4. The "new" target temperature of C1 is now 301.5°F. The resultant status of the problem is shown is Figure 6-8.

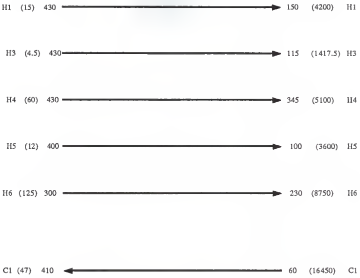


(a) grid diagram

Hot / Cold		H1	H2	H3	Hu	Qc
C1	$C1_1$	3675	---	---	8390	0
	$C1_2$	---	1540	---		
	$C1_3$	---	---	495		
Qh		0	0	0	0	0

(b) match matrix

Fig. 6-6. Hot subnetwork for 7SP4 problem: after stream splitting (step 4).

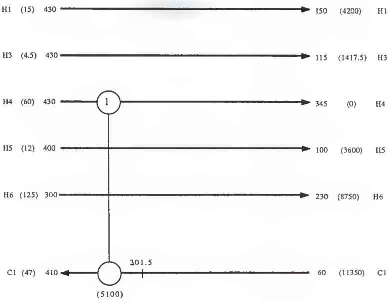


(a) grid diagram

Hot / Cold	H1	H3	H4	H5	H6	Qc
C1	* C	* C	H C	* C	* C	16450
CU	* C	* C	* C	* C	* C	6617.5
Qh	4200	1417.5	5100	3600	8750	23067.5

(b) match matrix

Fig. 6-7. Cold subnetwork for 7SP4 problem:
at the onset of synthesis (step 5).



(a) grid diagram

Hot Cold	H1	H3	H4	H5	H6	Qc
C1	* C	* C	5100	* C	* C	11350
CU	* C	* C	___	* C	* C	6617.5
Qh	4200	1417.5	0	3600	8750	17967.5

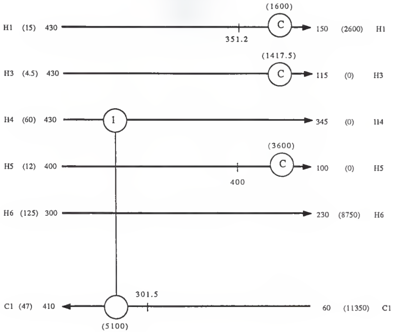
(b) match matrix

Fig. 6-8. Cold subnetwork for 7SP4 problem:
after the H4/C1 match (step 6).

Step 7. Next the cold utility can be matched with any of the four remaining hot streams, H1, H3, H5 and H6. Stream H5 is chosen since it has the lowest target temperature among all the hot streams. Higher target temperatures of the remaining hot streams imply increased ΔT_{ce} values; hence, they are likely to have "greater" feasibilities for subsequent matching. H5 gets eliminated and the heat load of Cu is reduced to 3017.5 Btu.

Step 8. The remaining part of Cu is matched with H3 because its target temperature is lower than that of the other two hot streams, H1 and H6. The match eliminates H3 and reduces the heat load of Cu to 1600 Btu.

Step 9. Cu is next matched with H1 since its target temperature is lower than that of H6. The match eliminates Cu and reduces the heat load of H1 to 2600 Btu. The "new" target temperature of H1 is 256.7°F. The resultant status of the solution is displayed in Figure 6-9. However, at this point, we can no match C1 with any hot stream at the hot end. We have arrived at a dead-end situation, thus necessitating the redistribution of cold utility. This time, instead of making the second match of Cu with H3, it is matched with H1. This match eliminates Cu and reduces the heat load of H1 to 1182.5 Btu. The new target temperature of H1 is 351.2°F. The resultant status of the solution is displayed in Figure 6-10. Now the H1/C1 match is feasible at the hot end.

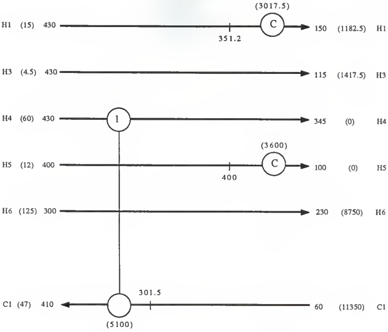


(a) grid diagram

Hot Cold	H1	H3	H4	H5	H6	Qc
C1	* C	---	5100	---	* C	11350
CU	1600	1417.5	---	3600	---	0
Qh	2600	0	0	0	8750	11350

(b) match matrix

Fig. 6-9. Cold subnetwork for 7SP4 problem: a dead-end situation after the H1/Cu, H3/Cu, and H5/Cu matches (step 9).



(a) grid diagram

Hot / Cold	H1	H3	H4	H5	H6	Qc
C1	H C	* C	5100	---	* C	11350
CU	3017.5	---	---	3600	---	0
Qh	1182.5	1417.5	0	0	8750	11350

(b) match matrix

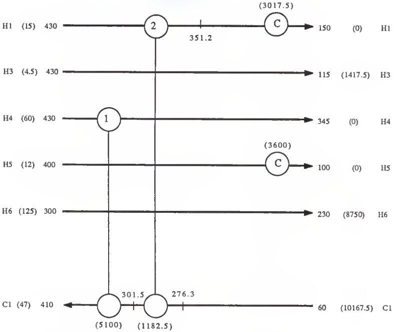
Fig. 6-10. Cold subnetwork for the 7SP4 problem: alternate arrangement for the coolers (step 9).

Step 10. Next, we generate H1/C1 match at the hot end, since no other hot steam can be used to heat C1 to its new target temperature of 301.5°F. The match eliminates H1 and reduces the heat load of C1 to 10167.5 Btu. The target temperature of C1 is changed to 276.3°F. The resultant status of the problem is depicted in Figure 6-11.

Step 11. Next, we make H6/C1 match at the hot end since the only other remaining hot stream, H3, can not be used to heat C1 to its largest temperature (the corresponding hot end match is not feasible). The match eliminates H6 and reduces tbe heat load of C1 to 1417.5 Btu. The "new" target temperature of C1 is 90.2°F. The resultant status of the problem is displayed in Figure 6-12.

Step 12. The last match is the H3/C1 match, which eliminates both the streams. Note that for elimination of both the streams in a match, it must be feasible at both ends and the two streams must have identical heat loads. The final solution for the cold subnetwork is shown in Figure 6-13.

Step 13. Combining the two subnetworks yields the overall solution to the 7SP4 problem, as shown in Figure 6-14.

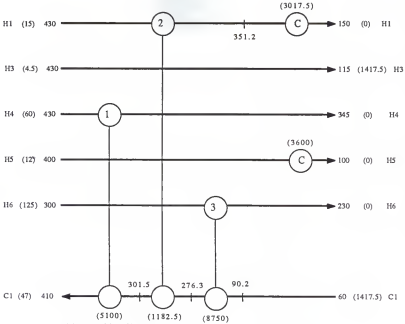


(a) grid diagram

Hot / Cold	H1	H3	H4	H5	H6	Qc
C1	1182.5	* C	5100	---	H C	10167.5
CU	3017.5	---	---	3600	---	0
Qh	0	1417.5	0	0	8750	10167.5

(b) match matrix

Fig. 6-11. Cold subnetwork for 7SP4 problem: after the H1/C1 match (step 10).

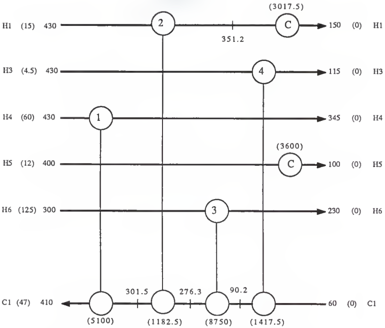


(a) grid diagram

Hot / Cold	H1	H3	H4	H5	H6	Qc
C1	1182.5	H C	5100	___	8750	1417.5
CU	3017.5	___	___	3600	___	0
Qh	0	1417.5	0	0	0	1417.5

(b) match matrix

Fig. 6-12. Cold subnetwork for 7SP4 problem: after the H6/C1 match (step 11).



(a) grid diagram

Hot / Cold	H1	H3	H4	H5	H6	Qc
C1	1182.5	1417.5	5100	---	8750	0
CU	3017.5	---	---	3600	---	0
Qh	0	0	0	0	0	0

(b) match matrix

Fig. 6-13. Cold subnetwork for 7SP4 problem: at the end of synthesis.

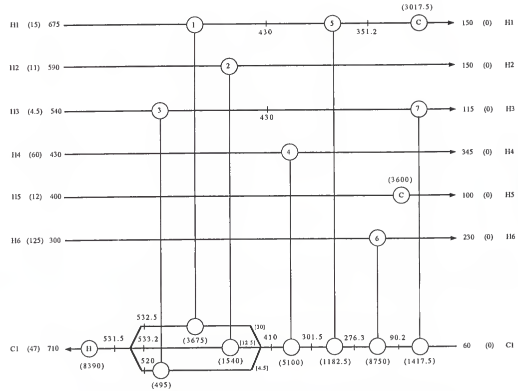


Fig. 6-14. 7SP4 problem: solution using the proposed approach.

Figure 6-15 shows the solution of the 7SP4 problem obtained by Papoulias and Grossmann (1982). Comparision of the present solution with it indicates that the former has only one instance of stream splitting whereas the latter has two instances of stream splitting. This can be attributed to the fact that the present approach does not resort to stream splitting until all possibilities of generating an unsplit solution have been explored. In other words, if it is possible to have a solution of a HEN problem without stream splitting, then the present approach will fd it; the more knowledge it contains, less amount of backtracking is required. In addition, the two solutions differ in the manner in which the splitting is performed. The stream splitting above the pinch point, common to both solutions, gives rise to different distribution of mc_p values. Consequently, the hot end temperatures of the candidate substreams, $C1_1$, $C1_2$ and $C1_3$, are different in the two solutions. For the present solution, these temperatures are very close to each other, within a range of 12.5°F, whereas for the other solution these temperatures are farther apart, having a range of 64.3°F. This difference arises due to the fact that the present approach attempts to generate split subnetworks with as near equal temperature drops across the candidate substreams as possible.

In summary, the proposed representation and manipulation schemes enable us to effectively utilize the available knowledge for synthesizing "better" HEN's. The success of these schemes is due to the rigorous formalization of the domain-knowledge necessary to solve the problem.

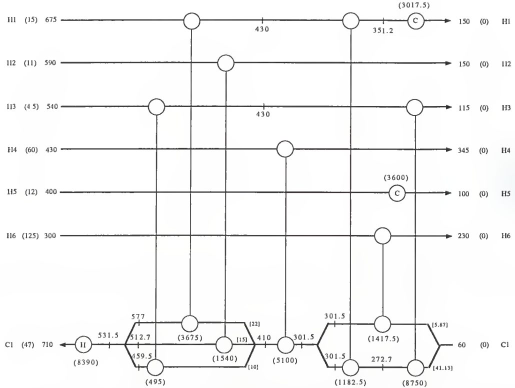


Fig. 6-15. 7SP4 problem: solution obtained by Papoulias and Grossmann (1982).

TOWARDS A KNOWLEDGE-BASED SYSTEM FOR HEN SYNTHESIS

The ultimate goal of the present work is to demonstrate the feasibility of process synthesis automation based on the knowledge-based approach. To accomplish this, we have chosen the problem of heat exchanger network synthesis. Up to this point, we have extracted and formalized some of the domain knowledge, and proposed representation and manipulation schemes to effectively utilize this knowledge. However, unless we ensure that the proposed strategy can be implemented on a machine, our demonstration is incomplete. Consequently, it is only fitting that we end this work with a brief discussion on how to transform the proposed strategy into a knowledge-based system.

Before attempting to develop a conceptual design of such a system, let us summarize some of the salient features of the proposed strategy that makes it a suitable candidate for our purpose.

(a) It enables us to synthesize HEN's featuring the minimum utility consumption and the minimum number of HTU's.

(b) At all times during the solution process, it displays the partial solution and the residual problem. This feature enables us to analyze and upgrade the strategy and/or the domain knowledge to improve the performance of the system.

(c) It shows explicitly all possible matches that can be made at any instant during the synthesis process.

(d) It can easily detect dead-end situations.

(e) It has an easy backtracking capability, enabling a system to explore "*what if ...?*" situations and to explain and justify the line of reasoning. In conjunction with (c) and (d), this feature ensures that if a solution exists, it can always be found.

(f) It is capable of identifying the situations warranting stream splitting, along with the participating streams.

(g) It can work with an incomplete and changing knowledge base, thus enabling us to build the knowledge base in an incremental fashion. This feature also allows us to explore various strategies for performing different subtasks, such as selection of streams for the "next" match, folding of a fully split pattern and the alternate match selection for backtracking.

None of the HEN synthesis methods proposed so far in the literature possesses even half of these features. In light of this, it is little wonder that we have not yet seen any computer-aided HEN synthesis system that is widely accepted by the industry or academia. The proposed system, described in the next few paragraphs, promises to reverse this trend.

The suggested architecture for the knowledge-based HEN synthesis system is the ***blackboard architecture*** (see, e.g., Reddy et al., 1976; Erman et al., 1980) which consists of a set of independent knowledge sources cooperatively solving a problem by communicating through a shared, common blackboard. This modular architecture allows a variety of combinations of knowledge sources and control strategies. It has been incorporated into the systems solving diverse tasks in crystallography, signal interpretation, vision, and psychological modeling (see, e.g., Barr and Feigenbaum, 1981). In contrast with these systems, the HEN synthesis system may involve knowledge sources that are not completely independent, but instead, have complex interrelations. This might entail modification of the classic blackboard architecture. The nature and type of modifications are implementational detail and need not be discussed at this point.

Figure 16 shows the conceptual design of the HEN synthesis system. The system has a blackboard architecture, with seven knowledge sources, and a user interface communicating with the blackboard. The user interface helps with problem specification and input/output of information between the user and the system. The blackboard contains the knowledge representation scheme for the problem, i.e., the grid diagram and the match matrix. It also contains the intermediate values and results that are required by various knowledge sources. Each knowledge source "reads" the blackboards and contributes towards the solution, based on the knowledge contained in it.

Each knowledge source (KS) deals with a particular aspect of HEN synthesis. Thus we have KS's corresponding to the selection of streams to match, matching, updating of the blackboard, backtracking, stream splitting, problem decomposition, and the explanation and reasoning. The knowledge within each KS can be partitioned depending upon the task it pertains to. Thus, the KS corresponding to stream splitting has partitions that deal with the generation of a split pattern, establishing the feasibility of a split pattern, folding of a split pattern, and generation of a split subnetwork. Similarly, backtracking consists of restoration of the blackboard, selection of the alternate matches and recognition of the situations requiring stream splitting. The tasks corresponding to each KS are recorded in Figure 6-16. The system performs the HEN synthesis task by following the stepwise procedure described in the preceeding section.

Note that the design of the knowledge-based HEN synthesis system proposed in this work is by no means in final form; this is the initial system configuration proposed at the outset of the implementation. As the conceptual and physical designs progress, the structure may change to a considerable extent.

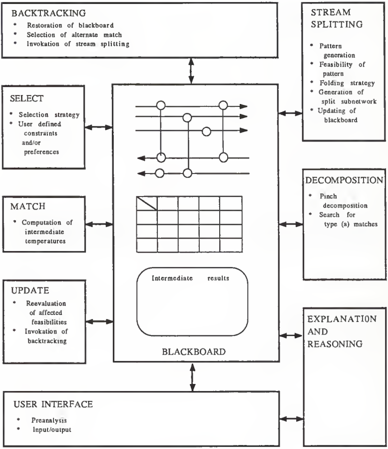


Fig. 6-16. Conceptual Design of a knowledge-based system for HEN synthesis.

CHAPTER 7. CONCLUSIONS AND RECOMMENDATIONS

Conventional approaches to computerize the synthesis task have not enjoyed much success. At the outset of the present work, therefore, a goal has been set to build an automated synthesis system, with a contention that a knowledge-based approach can provide us with a breakthrough in automating process synthesis. In the course of this work, a rationale for resorting to the knowledge-based approach has been presented. To begin with, the nature and scope of process synthesis have been examined; specifically, we saw how, in the past decade, the scope of synthesis has been narrowed down to exclude the task of conceptualization. This change of scope has resulted due to the limitation of the available computer technology. To include this task in an automated synthesis system, we need to borrow concepts from the latest developments in the field of Artificial Intelligence and Knowledge Engineering. Some of the Knowledge Engineering issues pertaining to process synthesis automation have been discussed.

Having seen the need for a knowledge-based approach, the remainder of the work dealt with demonstrating the approach using a specific synthesis subproblem, that of synthesizing heat exchanger networks. First, an exhaustive review of the available solution methodologies has been presented and the strengths and weaknesses of various approaches have been analyzed. Next, some of the knowledge required for synthesizing HEN's has been extracted and formalized. The major issue under consideration was how to conceive HEN's featuring the minimum utility consumption and the minimum number of units. In the process, the cause giving rise to the minimum number of units has been established; eventually, the elimination strategy has been conceived to attain it. To simultaneously attain both the optimality criteria, the effect of pinch point has been analyzed. Also, the necessary and sufficient conditions for the elimination strategy have been derived.

It is likely that the elimination strategy may lead to a dead-end situation, when it is not possible to continue HEN synthesis without violating at least one of the two constraints, the minimum utility and the minimum units. We saw how, in such situations, backtracking and/or stream splitting can be used to continue HEN synthesis without sacrificing the elimination strategy. In the process, a systematic procedure has been evolved to perform stream splitting by resorting to the elimination conditions derived earlier in the work.

Finally, the representation and manipulation schemes have been presented to effectively utilize the HEN synthesis knowledge that has been formalized in the present work. Based on these schemes, a systematic procedure has been proposed for HEN synthesis. The applicability and efficacy of the proposed method are demonstrated by solving the 7SP4 problem. This method, in turn, has given rise to the conceptual design of a knowledge-based system for HEN synthesis.

Having recapitulated the accomplishments of the present work, let us examine how far we have succeeded in attaining our goals. The systematic procedure proposed in this work generates HEN's featuring the minimum number of units, the minimum utility consumption, and minimal stream splitting. What has been obtained here is a computational model of the HEN synthesis process that closely mimics the reasoning pattern of an expert designer. This model will not only lead to an automated HEN synthesis system, but also enable us to experiment with novel ideas and strategies for various aspects of HEN synthesis, e.g., stream/stream match selection strategy that will minimize the backtracking, and incorporation of additional optimality criteria like minimizing the total heat transfer area of the network. The proposed knowledge-based system contains all the knowledge that has been formalized in this work and has provision for incorporating additional knowledge, as and when it is available in the required form. As a final note it is worthwhile to remember that in the process of building a computational model using the knowledge-based approach, we have obtained considerable insight towards solving the HEN synthesis problem in a "better" fashion, even without the use of computers. Generalizing this observation we can conclude that in attempting to automate the synthesis task, we will make ourselves better designers.

RECOMMENDATIONS FOR FUTURE WORK

There are two possible directions for continuing this work. One, we can proceed breadthwise in the synthesis field and use the knowledge-based approach to solve additional synthesis subproblems, including the separation system synthesis, reaction path synthesis, and control system synthesis. Once sufficient experience and insight have been obtained, then we can embark upon the all-encompassing problem of synthesizing the entire process flowsheets. However, before this can be accomplished, we need to build systems for the individual synthesis subproblems.

The second direction corresponds to the depthwise progress in the synthesis field. Additional knowledge for solving the HEN synthesis problem can be extracted, formalized, and incorporated in the proposed knowledge-based system. For this purpose, the following stepwise implementation scheme is suggested.

(a) As the first step, implement the core of the proposed knowledge-based system. This core can include the blackboard, the user interface, and the basic select-match-update cycle. In absence of any formal selection strategy, the selection of the pair of streams to be matched can be obtained from the user. The stream splitting can be restricted to the fully split pattern. Any folding, if required, can come from the user. Any backtracking in solving the problem can also come from the user. This skeleton or core system will provide us with an excellent tool for experimentation, to try out novel strategies.

(b) With the tool developed in (a), it will be possible to experiment with several strategies to evolve systematic procedures for folding during the stream splitting, selection of streams to match, backtracking, and identifying all possible matches of type (a) (i.e., matches that eliminate both the streams).

(c) With the additional knowledge obtained in (b), a full-fledged knowledge-based system, as proposed in the present work, can be developed. Additionally, the constraints arising from the resiliency, operability, and controllability of the resulting HEN can be readily added as separate knowledge sources. The resultant system will be capable of generating automatically an optimal HEN. With an explanation module, the system will also be able to supply the user the reasons and justifications for the results/decisions reached during the process of synthesizing a HEN.

LITERATURE CITED

1. 8oland, D., and Linhoff, B., "The Preliminary Design of Networks for Heat Exchange by Systematic Methods," *The Chemical Engineer*, 222-228, April (1979).

2. 8arr, A., and Feigenbaum, E. A., *The Handbook of Artificial Intelligence, vol I*, William Kaufmann, Inc., Los Altos, CA (1981).

3. Cerda, J. and Westerberg, "Minimum Utility Usage in Heat Exchanger Network Synthesis - A Transportation Problem," ORC Report No. 06-16-80, Carnegie-Mellon University, Pittsburgh, PA (1980).

4. Erman, L. D., Hays-Roth, F., Lesser, V. R. and Reddy, O. R., "The HEARSAY-II Speach Understanding System: Integrating Knowledge to Resolve Uncertainty," Computing Surveys, 12 (2), 213-253 (1980).

5. Flower, J. R., and Linhoff, B., "Thermodynamic Analysis in the Design of Process Networks," CACE `79, April 6-11, Montreux, Switzerland (1979).

6. Flower, J. R., and Linhoff, 8., "A Thermodynamic - Combinatorial Approach to the Design of Optimum Heat Exchanger Network," AIChE J., 26 (1), (1980).

7. Grennkom, R. A., Koppel, L. B., and Raghawan S., "Heat Exchanger Network Synthesis - A Thermodynamic Approach," 71st AIChE Meeting, Miami, FL (1978).

8. Grossmann, I. E., and Sargent, R. W. H., "Optimum Design of Heat Exchanger Networks," Computer and Chem. Eng., 2 (1), (1978).

9. Hendry, J. E., Rudd, D. F., and Seader, J. D., "Snthesis in the Design of Chemical Processes," AIChE J., 19 (1), 1-15, (1973).

10. Hohmann, E. C., "Optimal Networks for Heat Exchange," Ph.D. Thesis, Univ. S. Calif. (1971).

11. Hwa, C. S., "Mathematical Formulation and Optimization of Heat Exchange Network Using Separable Programming," AIChE-Intern. Chem. Eng. Symp. Series No. 4, 101 (1965).

12. Linhoff, 8., and Flower, J. R., "Synthesis of Heat Exchanger Networks, Part I. Systematic Generation of Energy Optimal Networks," AIChE J., 24, 633 (1978a).

13. Linhoff, 8., and Flower, J. R., "Synthesis of Heat Exchanger Networks, Part II. Evolutionary Generation of Networks with Various Criteria of Optimality," AIChE J., 24, 642 (1978b).

14. Linhoff, 8., "Thermodynamic Analysis in the Design of Process Networks," Ph.D. Thesis, University of Leeds, England (1979).

15. Linhoff, B. and Hindmarsh, E.; "The Pinch Design Method of Heat Exchanger Networks,"

16. Linhoff, B., Townsend, D. W., Boland, D., Hewitt, G. F., Thomas, B. E. A., Guy, A. R. and Marsland, R. H., *A User Guide on Process Integration for the Efficient Use of Energy*, The Institute of Chemical Engineers, Rugby, England (1982).

17. Motard, R. L., and Westerberg, A. W., *Process Synthesis*, AIChE Advanced Seminar Lecture Notes, New York (1978).

18. Nishida, N., Kobayashi, S., and Ichikawa, A., "Optimal Synthesis of Heat Exchange Systems - Necessary Conditions for Minimum Heat Transfer Area and their Applications to System Synthesis," Chem. Eng. Sci. 27, 1408 (1971).

19. Nishida, N., Stephanopoulos, G., and Westerberg, A. W., "A Review of Process Synthesis" AIChE J., 27, 321 (1981).

20. Papoulias, S. A. and Grossmann, I. E., "A Structural Optimization Approach in Process Synthesis - II. Heat Recovary Networks," Comput. & Chem. Eng., 7 (6), 707-721 (1983).

21. Pehler, F. A. and Liu, Y. A., "Studies in Chemical Process Design and Synthesis: VI. A Thermoeconomic Approach to the evolutionary synthesis of Heat Exchanger Networks," Chem. Eng. Comm., 25, 295 (1984).

22. Pho, T. K. and Lapidus, L., "Topics in Computer-Aided Design II. Synthesis of Optimal Heat Exchanger Networks by Tree search Algorithms," AIChE J., 19, 1182 (1973).

23. Rathore, R. N. S., and G. J. Powers, "A Forward Branching Scheme for the Synthesis of Energy Recovery Systems," Ind. Eng. Chem. Process Design & Development, 14, 175 (1975).

24. Reddy, D. R., Erman, L. D., Fennell, R. D. and Neely, R., "The HEARSAY-II Speach Understanding System: An example of the Recognition Process," IEEE Trans. Comp. C-25, 427-431 (1976).

25. Rich, E., *Artificial Intelligence*, McGraw-Hill, New York (1983).

26. Rudd, D. F., Powers, G. J. and Seader, J. D., "Synthesis in the Design of Chemical Processes," AIChE J., 19 (1), 1-15 (1973).

27. Siirola, J. J., "Chemical Process Synthesis," Chem. Eng. Education, XVI (2), 68-71 (1982).

28. Siirola, J. J. and Rudd, D. F., "Computer-Aided Synthesis of Chemical Process Designs," Ind. Eng. Chem. Fund., 10 (3), 353-362 (1971).

29. Sowa, J. F., *Conceptual Structures: Information Processing in Mind and Machines*, Addison Wesley, New York (1984).

30. Stefik, M. and Bobrow, D. G., "Object Oriented Programming: Themes an Variations," AI Magazine, 6 (4), 40-62 (1986).

31. Umeda, T., "Computer-Aided Process Synthesis," Comput. & Chem. Eng., 7 (4), 279-309 (1983).

32. Umeda, T., Harada, T., and Shiroko, K., "A Thermodynamic Approach to the Syntheisis of Heat Integration Systems in Chemical Processes," Proc. 12th Symposium on Computer Applications in Chemical Engineering, p. 187, Montreux, Switzerland (1979).

33. Umeda, T., Itoh, J., and Shiroko, K., "Heat Exchange System Synthesis," Chem. Eng. Prog., 74, 70 (1978).

34. Umeda, T., Niida, K., and Shiroko, K., "A Thermodynamic Approach to Heat integrationin Distillation Systems," AIChE J., 25, 423 (1979).

35. Westerberg, A. W., "A Review of Process Synthesis," ACS Symposium Series 124 (1980).

KNOWLEDGE-BASED APPROACH FOR PROCESS SYNTHESIS AUTOMATION:
AN APPLICATION TO HEAT EXCHANGER NETWORK SYNTHESIS.

by

CHETAN MEHTA

B. Tech., Indian Institute of Technology, Bombay, 1982.

AN ABSRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Chemical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1986

ABSTRACT

Process synthesis is one stage of design where efforts at computerization have not enjoyed much success. In this work a knowledge-based approach to process synthesis is presented. A knowledge-based approach is comprised of extraction, formalization, representation and manipulation of domain-specific knowledge to solve complex problems. The need and rationale for resorting to such an approach to develop automated process synthesis systems are discussed. Some Knowledge Engineering issues relevant to process synthesis automation are also discussed, along with the desired characteristics of the resultant systems. The approach is demonstrated by applying it to the wellknown problem of Heat Exchanger Network (HEN) synthesis. To begin with, a comprehensive review of the nature and scope of the existing HEN synthesis methodologies is presented. Strength and weaknesses of the currently employed approaches are briefly discussed.

As the first step of the knowledge-based approach, HEN synthesis knowledge is formalized. The cause leading to the minimum number of units for a HEN is established. The effect of a pinch point is analyzed and an elimination strategy is proposed to attain the minimum number of units for a HEN. The necessary and sufficient conditions for adhering to the elimination strategy are derived. The need for stream splitting, to generate HEN's featuring the minimum number of units and the minimum utility consumption, is established with the help of an illustrative example. A systematic procedure to carry out the stream splitting in accordance with the elimination strategy is developed.

As the second step of the knowledge-based approach, schemes for representing and manipulating the HEN synthesis knowledge are proposed. A stepwise procedure that employs the the proposed representation and manipulation schemes, is proposed for HEN synthesis. The procedure generates HEN's with the minimum number of units, the minimum utility requirement and the minimal amount of stream splitting. Based on this procedure, a conceptual design of a knowledge-based system for HEN synthesis is presented, along with its desired characteristics. An incremental implementation strategy is recommended for constructing the system.