Constraint Logic Programming for Chemical Process Synthesis

Hussein A. Abbass*; Geraint A. Wiggins†; Ramachandran Lakshmanan‡; Bill Morton‡
$\star$ School of Computing Science, QUT, Australia[1]
† Division of Informatics, University of Edinburgh, Scotland
‡ School of Chemical Engineering, University of Edinburgh, Scotland

Abstract

We present a successful application of Constraint Logic Programming (CLP) to a major problem in Chemical Process Synthesis: the retrofit of heat exchanger networks. The method combines expert intuition, represented in a set of heuristics, with linear programming, an optimisation technique, by means of a program in the CLP paradigm. The results shown are mostly significantly superior to those reported in the chemical process synthesis literature. The example presented here shows how a company could save $35,920 per year while incurring less than half the capital costs of other published solutions.

The method is novel in that it uses a combination of mathematical optimisation techniques with backtracking heuristic search to achieve its results.

1 Introduction

The integration of Constraint Programming (CP) technology with Knowledge Based Systems (KBS) naturally and inevitably exposes CP to wider domains of application. Real life applications are often complex and messy with much obscure embedded knowledge and information. Domain dependent techniques are vital, if solutions are to be realistic and reachable. In this paper, we illustrate how we use domain specific knowledge in a complex domain – chemical process synthesis – to reduce the complexity of the problem so that it becomes feasible for CP techniques to tackle and solve it. Moreover, the resulting algorithm produces better solutions than existing approaches – sometimes better than purely mathematical optimisation techniques, which ignore domain specific knowledge, themselves.

Process Synthesis (Nishida, Stephanopoulos, and Westerberg 1981) involves the construction and interconnection of unit operations required for the conversion of the raw materials of a chemical process into its finished products. Traditionally, process synthesis consists of three main entities: reaction of chemicals, separation of the results and energy integration (sources of external heat or cooling). In traditional approaches to process design, the reaction stage defines the tasks for the subsequent design of the separators. The utility requirements for the reaction and separation stage are determined last, heating and cooling being regarded as auxiliary operations in the process. However, this naïve approach can negatively affect the total operational cost of the plant.

Before the petroleum crises in the 1970s, energy costs usually represented around 5% of the total plant cost. Subsequently, the energy cost component rose to around 20%, causing the industry to rethink its approach to process design in more parsimonious terms. Since then, the problem of the

---

[1] Correspondence should be addressed to the first author, at the Data Mining Lab, Machine Learning Research Centre, Queensland University of Technology, GPO Box 2434, Qld4001, Brisbane, Australia. Email: abbass@fit.qut.edu.au. Web: http://www.fit.qut.edu.au/~abbass/

design of *Heat Exchanger Networks* (HENs) – one of the main process synthesis problems – has been receiving a great deal of attention.

The rest of this paper is structured as follows. Section 2 gives a brief sketch of the background of HEN design. In Section 3, different approaches for design and retrofitting HENs are briefly introduced. The proposed method is discussed in Section 4 followed by a case-study in Section 5, and then conclusions are drawn in Section 6.

2 Description of Problem Domain

Heat exchanger networks utilise the energy content of streams of hot fluid to heat cold ones. The transfer of energy from the hot stream to the cold stream depends on the rate of flow, the area of the exchanger, the heat transfer coefficient, the temperature gradient along each stream, and other factors. The advantage of HENs is that they allow the recycling of energy by taking it out of hot products and passing it into cold raw materials which required heating, and *vice versa*. Sometimes, it is necessary to supplement the function of a HEN with direct cooling or heating from external sources. These external sources are called *utilities.*

Much research has been published about the initial or *grass-roots* design of HENs, where a brand new HEN is being set up. Gundersen and Naess (1988) give a comprehensive review of the field. In a typical grass-roots problem, the requirement is to find the optimal design that minimises the utility cost and exchanger network area, and to select appropriate network topology for the new network.

Typical examples of constraints in grass-roots HEN design are the first and second law of thermodynamics. Roughly speaking, the first law is concerned with the energy balance (*i.e.*, the *conservation of energy* constraint which guarantees that the energy input is equal to the output less any loss or plus any gain). The second law requires that the temperature of a hot stream is always greater than the temperature of the cold stream to which it is transferring heat. It follows from this that a practically useful constraint is one which ensures *efficient* exchange, the difference between the two temperatures must not be less than a specific value called the *exchanger minimum approach temperature* (EMAT). In some cases (*e.g.*, where materials are chemically incompatible) a *nomatch* constraint may apply (*i.e.* a match between two particular streams or one stream and a particular exchanger is forbidden).

A more difficult case involves the *Retrofit of Heat Exchanger Networks.* Here, the problem is more complicated as a network exists already, and the question is how to reconfigure the current equipment either to achieve new objectives, or to give a better solution to the existing ones. The retrofit problem is hard because one is restricted, where possible, to use the available heat exchangers and matches, and to minimise the amount of change. However, this cannot be done naïvely, because the obvious solution is not always the best.

So the designer in a retrofit problem is constrained not only with all the constraints involved in a grass-roots problem, but also with legacy constraints as well.

The existing exchangers represent constraints on the problem as it is uneconomic and ecologically unfriendly to get rid of them. The exchangers impose implicit constraints on the problem, such as the existing area, the type of materials, and the existing matches. Rough heuristic methods exist for reasoning about the cost of retrofit, but the problem is essentially non-linear, which results in inability of most traditional optimisation techniques to handle the problem in real life. The search space grows exponentially with the number of exchangers in the network, which denies us the possibility of exhaustive search without an effective way to prune the search space. These last two points will be explained in detail below, when we introduce our method.

The solution of the grass-roots HEN problem is addressed in the literature by means of two main approaches. In the first, the problem is solved by *pinch technology* introduced by Linnhoff and Hindmarsh (1983). This method is based on thermodynamics and achieved some success in grass-roots design – in spite of its major drawback of not reaching the optimal solution. In the second popular approach, *optimisation methods* are applied to grass-roots HEN design – regardless of the high computation costs of reaching the optimal solution and the possibility of getting stuck in a local optimum. Methods have been developed to overcome these drawbacks by decomposing the solution steps (Yee et al. 1990; Yee and Grossmann 1990).

The harder problem of HEN retrofit is solved using a combination of pinch targeting and optimisation models (Yee and Grossmann 1991; Briones and Kokossis 1996). This combination aims to overcome the burden of computation in optimisation models by developing some initial boundaries using the pinch method.

Genetic algorithms have been used as a powerful search mechanism (Androulakis and Venkatasubramanian 1991) though, in this instance, the fitness function was so inefficient as to be impractical, involving non-linear constraint solution for each evaluation. A more practical approach is described in (Lewin, Wang, and Shalev 1998; Lewin 1998).

Although genetic technology does not guarantee the global optimum, with more complicated problems it can result in a robust solution.

Another method, discussed by Tjoe and Linnhoff (1986) and further implemented by Lakshmanan and Bañares-Alcántara (1996), is *retrofit by inspection.* Retrofit by inspection is heuristic search, based on the engineer's intuitions, for better structures. However, this method does not guarantee an optimal answer in real life problems, because it is impossible to optimise the structure in full.

Although the HEN problem has received much attention in the literature, the problems are often presented in ways suited to the algorithm which each paper suggests. We have tried to be more general than this when specifying our algorithm.

3 Approaches for solving HEN retrofit

Research in HEN retrofit may be divided into four main approaches, though there is some cross-fertilisation between them. There follows is a brief summary for each of the four, but for a comprehensive discussion, the reader is referred to the old but good review paper of Gundersen and Naess (1988).

3.1 Pinch technology

The idea of the *pinch point* was introduced by Umeda et al. (1979) and further formalised by Linnhoff and Hindmarsh (1983). The pinch point (Shokoya 1992), where the minimum temperature difference between two streams is observed (*i.e.*, the difference between the hot temperature and the cold temperature is equal to the Heat Recovery Approach Temperature (HRAT), which results from establishing a tradeoff between energy and area using the Pinch approach), determines the degree of possible heat recovery. The pinch divides the process into two energy subsystems; above the pinch and below it, each of which is in enthalpy balance. Thermodynamically, it is not efficient to transfer heat across the pinch (*i.e.,* from a hot component over the pinch to a cold component below the pinch). The transfer of heat across the pinch is called *cross-pinch heat exchange* (sometimes inaccurately called *criss-cross*) and is considered by HEN designers to be a loss in energy. Cross-pinch heat exchange is not recommended in grass-roots HEN design, but in retrofit it is sometimes necessary to transfer heat across the pinch because of legacy constraints.

Although pinch technologies have been successfully applied to grass-roots design of HENs, they have not proven to work well with HEN retrofit (Yee and Grossmann 1991). One of the credits to pinch technology, in general, is the provision of good estimates for determining the targets (*e.g.*, upper bounds on additional exchangers). In grass-roots problems, the pinch technology has succeeded in doing so, whereas in HEN retrofit it has failed in some important cases (Lakshmanan and Bañares-Alcántara 1996).

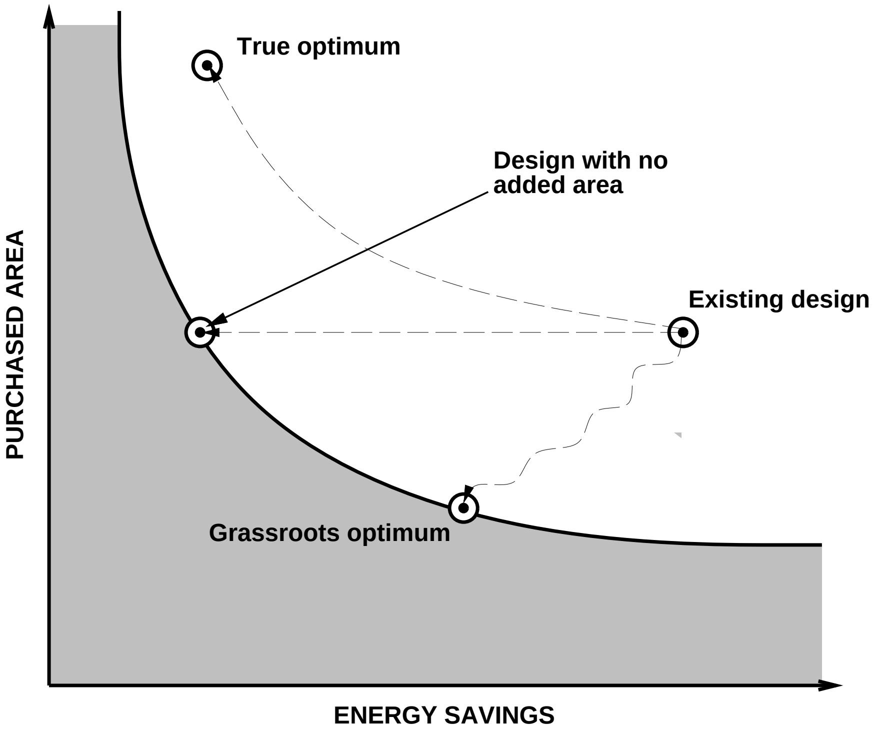


Figure 1: Area targeting trade-off

Pinch technology works by establishing a tradeoff between energy savings and capital expenditure in the form of purchased exchanger area. (See Figure 1). The curve of the graph represents the thermodynamically feasible limits of design. It is impossible to create a HEN which lies in the shaded region of the figure. Most existing designs will lie somewhere to the right and above the curve, such as the point shown. Clearly, such a design is less than efficient since it requires area greater than the thermodynamic minimum for the energy savings represented by the x-coordinate of the point. For a grassroots design, the optimum lies on the curve as shown. When carrying out a retrofit, it is tempting to try and follow a path shown by the zig-zag arrow, aiming to reach the optimum grassroots design. However, the grassroots optimum has a smaller amount of purchased area, so the zig-zag path implies that existing area in the HEN is to be disposed of. Since this area has already been paid for, the path to the grassroots design is clearly non-optimal. On the other hand, one might attempt to use *only* the existing area and attempt to improve the efficiency of the HEN solely through topology modifications (represented by the horizontal path to the left). In general, however, since the existing area is available in discrete “chunks,” it is physically impossible to follow this path. The real optimum, therefore, involves the combination of topology modifications with the purchase of additional area, following the curved path shown in the diagram.

3.2 Optimisation

A lot of work has been conducted in formulating a mathematical programming model of HENs. The grass-roots problem has been solved by a number of researchers such as Yee et al. (1990), Yee and Grossmann (1990), Yee, Grossmann, and Kravanja (1990). The problem of retrofit has also been tackled. In general, mathematical models have the advantage of finding optimal solutions. But the usual disadvantages of mathematical models – such as, the inability to represent qualitative constraints like flexibility, the assumption that the model works in the steady state, the difficulties raised from the complexity of the problem (*e.g.*, nonlinearity and number of variables and equations), and the inability to report the rationale behind the solution – apply in HEN design as elsewhere.

The first approaches, such as those of Kelser and Parker (1969) and Kobayashe, Umeda, and Ichikawa (1971), concentrated on the formulation of the problem as an assignment problem. More recent attempts have included more detailed models of the problem, and as a result the model becomes more complicated and less computationally tractable (Viswanathan and Evans 1987; Yee et al. 1990; Yee and Grossmann 1990; Yee, Grossmann, and Kravanja 1990).

In HEN retrofit, a considerable amount of work has been done to formulate the problem as a mathematical programming model. Yee and Grossmann (1991) introduced a Mixed Integer Non-Linear Programming (MINLP) formulation. They examined a number of optimisation methods involving approximations of upper and lower bounds: outer approximation/equality relaxation (OA/ER) with piecewise approximations; generalised benders decomposition with valid outer approximations; and augmented penalty version of the OA/ER. With any of these optimisation methods, the MINLP problem is divided into a master mixed integer linear programming (MILP) with nonlinear programming (NLP) subproblems. The NLP subproblem optimises a particular network structure by fixing the binary variables and yielding an upper bound to the cost. The MILP master problem is optimised with an approximated feasible region to select new network structures and to predict lower bounds on the cost. The problem with these methods is that they have been shown empirically to be size-limited.

Briones and Kokossis (1996) introduced an algorithm that addresses the problem as a multi-task effort and applies a decomposition scheme which uses both mathematical programming and pinch analysis methods. The different tasks are targets for structural modifications and heat transfer area changes, the development and optimisation of the retrofitted network and the analysis of its complexity against economic penalties and trade-offs. The decomposition stages embed targeting information which supports screening and facilitates an effective optimisation search. As such, the decomposition not only bypasses the limitations of past decomposition techniques but exploits its features toward the development of an interactive design tool.

Asante and Zhu (1997) have introduced a new decomposition approach for solving the HEN retrofit problem. The decomposition starts by searching for topology changes using a set of mixed integer linear programming (MILP) models followed by a heat recovery and area optimisation stage using a NLP model. From our practical point of view, Asante and Zhu's assumption of omitting cost data in their objective function is unacceptable, and their argument that the absolute optimality of a design in terms of both cost and practical considerations can not be guaranteed is flawed. Indeed, we suggest that the omissions would not be acceptable even if the argument were valid. The user always looks for a compromise between different objectives when a conflict exists. Further, repiping across different streams is not considered – greatly reducing the search space – and consequently giving suboptimal results.

3.3 AI approaches to HEN design

Knowledge based systems were mainly introduced in HEN design to overcome the difficulties with the mathematical model and sometimes to reduce the complexity of the problem to be solved later by mathematical programming methods. A considerable number of Artificial Intelligence (AI) techniques have been applied to grass-roots HEN design, but none has been tried on the retrofit problem. Expert systems and genetic algorithms were also applied to grass-roots design (Garrard 1996; Androulakis and Venkatasubramanian 1991; Wang, Shalev, and Lewin 1997).

3.4 Retrofit by inspection

Inspection methods are designed mainly for HEN retrofit problems rather than grass-roots design. They have not always been considered good solutions: Tjoe and Linnhoff (1986) state that inspection could never work, and so argue the need for the Pinch method. Lakshmanan and Bañares-Alcántara (1996) have developed a visualisation tool to support the engineer in the retrofit by inspection approach, and proposed some general guidelines to help the engineer in spotting the needed changes. One of the main contributions of Lakshmanan and Bañares-Alcántara's method is the introduction of the *retrofit thermodynamic diagram* which visualises the criss-cross in the network. The main disadvantages of the method are:

1. The diagram does not guarantee to show all criss-crosses in the network. Also, the diagram may show a criss-cross, where no criss-cross exists.
2. The criss-cross visualised on the screen does not have an explicit relation to the degree of criss-cross.
3. The guidelines are addressed in a general form which need to be more formal if they are to be automated.
4. The user has to check the *exchanger approach temperature* (EMAT) herself from a spreadsheet. The EMAT is not explicitly shown in the diagram.
5. The visualisation tool does not visualise stream splitting or phase change.

We have formalised the guidelines of Lakshmanan and Bañares-Alcántara, and used them in our system. They have demonstrated that their guidelines resulted in better solutions than previous attempts in some cases and this suggests promise in the retrofit by inspection method.

4 ITRI: A New Approach to Automating HEN Retrofit

The knowledge base for our system, called ITRI[1], has three main components, obtained from literature surveys and from a domain expert. The first main component concerns network optimisation. This is mainly for generating the initial mathematical representation of the model. The second concerns the problem domain. Expert opinion was elicited into a set of heuristic rules to guide search within the system. The final component is a suitable knowledge representation.

---

[1] Intelligent Tool for Retrofit by Inspection

4.1 Knowledge acquisition

In order to elicit the knowledge required for our system, a number of meetings were conducted with the domain expert. In the first couple of meetings, the problem domain was introduced, by means of a demonstration of the package that the domain expert has developed. This was a useful starting point, allowing him to talk about his own work and its evaluation, and allowing us to understand his expectations from the developed system. Subsequently, we prepared a series of questions before each meeting, which led directly into discovering the necessary heuristics.

It is worth mentioning that the nature of the problem is different from considering the program as an expert system. The knowledge that we have acquired from the domain expert is used not only to build the heuristic set of rules, but often also to exclude unnecessary rules that are infeasible and redundant though they seemed initially feasible. As an example for this last point, one of the rules that seemed to be feasible is to forbid the repiping after the addition of a new exchanger as the system will force the new exchanger to be placed in the right position in the network. This seemed logical though he has mentioned that the repiping might be necessary after adding a new exchanger to utilise an unutilised area in one of the exchangers in the network. This point in our system resulted in having a repiping check after each change in the network. Also, the knowledge conducted from the domain expert played the major rôle in formalising the problem.

A full description of our formalisation of the HEN Retrofit problem is given in Appendix A.

4.2 Problem representation

The heat exchanger network is represented as a sequence of the form

$$M_{ex} = \{M_1(Type, Area, \alpha), \ldots\ldots, M_k(Type, Area, \alpha)\}$$

where $Type$, is the type of match that the exchanger matches (“he” for process-process match, “hu”, for process-hot utility match, and “cu” for process-cold utility match), $Area$ is the area of the exchanger, and α is the amount of energy loss corresponding to the material of the exchanger. Each M_i term represents an exchanger unit in the network.

The current streams in the network are represented using two further sequences, one for the cold streams and another for the hot ones. The elements of each of these are tuples of variables defining the characteristics of the stream and the set of exchangers connected to this stream. The sequence of hot streams takes the form

$$HS = \{H_1(T_{in_{H_1}}, T_{out_{H_1}}, HMCP_1, , Eseq(H_1)), \ldots, H_n(T_{in_{H_n}}, T_{out_{H_n}}, HMCP_n, Eseq(H_n))\}$$

where $T_{in_{H_i}}$ and $T_{out_{H_i}}$ represent the inlet and outlet temperatures for hot stream H_i respectively, $HMCP_i$ represents the material coefficient of the i's hot stream, and $Eseq(H_i)$ is a sequence of all sets of exchangers on hot stream i. Each element in this sequence takes the form

$$\begin{aligned}&\langle \mathit{Exchanger\ label},\\ &\ \mathit{Predecessor\ list},\\ &\ \mathit{Successor\ list},\\ &\ \mathit{Inlet\ temperature},\\ &\ \mathit{Outlet\ temperature},\\ &\ \mathit{Load},\\ &\ \mathit{Area},\\ &\ \mathit{Mcp}\rangle\end{aligned}$$

The type of costs involved in the decision are: moving a heat exchanger; re-piping a stream; purchasing a new heat exchanger; additional exchanger area; hot and cold utilities.

4.3 The Search Mechanism

ITRI solves the HEN retrofit problem by applying four groups of rules recursively while optimising the loads on the utilities. To control the combinatorial explosion thus produced, some heuristics are applied. The algorithm is outlined in Figure 2.

```
Procedure Optimise
        * Construct a linear programming model.
        * Optimise the model.
Procedure Reallocate
        * Do
            - Apply the exchanger-reallocation rule.
            - Call procedure Optimise.
            - Evaluate using the total cost.
        * While there is no further improvement.
Procedure Split
        * Do
            - Apply the Split rule.
            - Call procedure Optimise.
            - Evaluate using the total cost.
        * Until there is no further improvement.
Procedure Add New
        * Do
            - Apply the adding-new-exchanger rule.
            - Call procedure Optimise.
            - Evaluate using the total cost.
        * Until there is no further improvement.
Main Loop
        * Call procedure Optimise.
        Label A:
        * Call procedure Reallocate.
        * Call procedure Split.
        * If Split did not improve, Call procedure Add New.
            - Else Go to A.
        * If Add New did not improve Stop.
            - Else Go to A.
```

Figure 2: The ITRI algorithm

4.3.1 Heuristics

Optimisation of Objective Function The first heuristic aims to optimise the objective function. The overall objective of the problem in our formulation is to minimise both the energy cost and the cost needed for the changes. The problem involves both linear and nonlinear costs. The linear cost is used during the optimisation while the nonlinear cost is used to prune the search space. The sequence of events is as follows: a suggestion is made by the heuristic rules followed by optimising the network using the linear part of the objective then evaluating the solution using the total objective (*i.e.* linear and nonlinear). If the solution results in a payback period more than three times the payback period required by the user, the algorithm refuses the solution and backtracks. The "three times the user payback period" is a rule of thumb for setting an upper bound for the payback, which stems from the fact that the cost varies non-monotonically with the progress of the search. This would be the case even if there were no non-linear costs, since each modification is changing the network structurally and we have no way of proving monotonicity. The total cost might increase after the nonlinear part is added then after an iteration or two it starts decreasing again. The factor three was suggested by the domain expert to be a reasonable limit to ensure that potentially good solutions were not screened out early.

Decreasing Loads on Utilities The second heuristic imposes an order on the search space: if a given HEN reconstruction step does not result in a decrease in the utilities' loads, the step is rejected and the algorithm backtracks, trying to find a different change. It might be supposed that this heuristic would suffer badly from a horizon effect, but it does not. For example, suppose we need to do a repiping (which will increase the load on a utility) for the sake of adding an exchanger later on. By rejecting the repiping step, the step is still possible because the algorithm tries to add an exchanger anyway and after that it cycles again, first trying the repiping. So in fact, what is constrained here is the order that the process follows, rather than the final result. However, importantly, the number of evaluations is reduced.

Adding Exchangers In the third heuristic, it is assumed that the addition of an exchanger will be valid only when the exchanger is needed to create a path between a hot and a cold utility which will result in a reduction in the total utility loads. This is the most significant flaw in our algorithm, because, in a few cases, we need to add more than one exchanger to create the path, which this heuristic prevents. However, it works well in most cases. Indeed, there is only one published solution (of more than 10) in which more than one exchanger was needed to create a path.

Occurrence Checks In addition to the heuristics that are used to reason about the search space, a number of occurrence checks have been included to optimise the structure of the network as it is constructed.

The first check is whether the load on one of the utility exchangers has reached zero. If so, the algorithm chooses the newly added exchanger with the highest load to be replaced by this utility exchanger.

The second is a check on the payback of the current solution. If the current payback is less than the best payback found so far, the current solution is reported to the user. This is very useful when the search space is very large: it enables the user to follow the progress of the search and to impose an early termination if she is satisfied with the current solution.

The third check simply avoids loops within the heat path.

The last check tests whether the CLP(R) library used in the implementation returns multi-optimal solutions. In this case, the algorithm creates a choice point and replaces the bindings of the variables with their infimum and supremum.

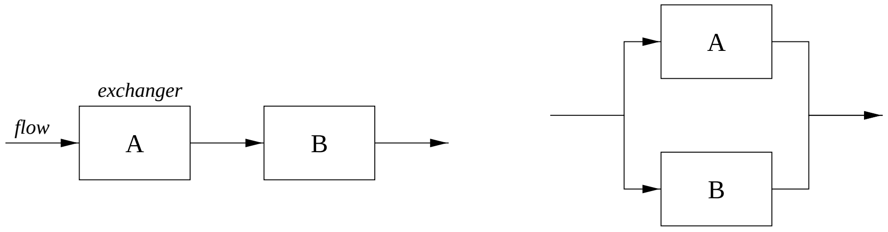
4.3.2 The Four Main Rule Groups

The following four groups of rules encode the majority of the domain knowledge used in the ITRI system.

Load shifting Load shifting, the moving of a flow from one exchanger to another or from utilities to process exchangers, is conducted at two points: before doing any change in the network; and after each change to the network. Load shifting is done by the linear programming model which guarantees that the maximum load to be shifted in the network as a whole will be reached and accordingly the maximum heat recovery will be achieved. The model achieves this by minimising the total utility cost, which results also in the minimum total utility loads. This has the advantage of not carrying the load shifting at a time and the disadvantage of skipping an optimal solution if the optimality is not at the maximum heat recovery.

Exchanger reallocation Within a Heat Exchanger Network, the most efficient use of area is made when the heat transfer is "vertical". This means that the hottest parts of the hot streams exchange heat with the hottest part of the cold streams and the colder parts of the hot streams exchange with the colder parts of the cold streams. When this is not the case, we say that *Criss-cross* heat exchange exists. Possibly the most important goal of a HEN retrofit study is to eliminate or reduce criss-cross which is normally done by swapping either the hot or cold sides of the two exchangers involved. Criss-cross is detected as follows. Suppose that we have two exchangers, A and B, in the network. Assume that the inlet hot temperature for B is greater than the inlet hot temperature for A. If the inlet cold temperature for A is greater than the inlet cold temperature for B, then this constitutes a criss-cross, so the heat exchange is not as efficient as it should be. This can be corrected by swapping two exchangers. After interchanging the two exchangers, load shifting is performed and the feasibility of the network is checked. If the network has become infeasible, the reallocation is rejected and the algorithm, by backtracking, tries to find another two exchangers to be reversed. The algorithm continues the reallocation until no other useful interchanging in the network exists. Only the exchangers on the hot side need to be tested. Moreover, because the exchangers are checked pairwise, the complexity of the check has an upper bound of the square of the number of exchangers – which is polynomial and accordingly computationally cheap, so the algorithm is efficient. Using this algorithm, no criss-cross in the network can be missed, and no false criss-cross can be detected.

Stream splitting The idea of stream splitting – dividing an existing stream between two exchangers, and thus perhaps using it more efficiently – is formalised by generating the set of *bottlenecks* in the network. The configuration resulting from splitting a stream flowing through two exchangers is shown in Figure 3. An exchanger creates a bottleneck if it prevents a load from being shifted on a load path. This case can be easily detected, because we shift the maximum load possible each time. After shifting the maximum load, the exchangers in the network are tested against the second law of thermodynamics. If an exchanger has a temperature difference equal to its minimum approach temperature, then either this exchanger prevents a load from being transferred or it has no effect. In either case, an attempt is made to split the stream before the offending exchanger and placing that exchanger in parallel with one of its neighbours, on either the hot or the cold side or both. If the splitting results in no improvement in the total utility load or it results in an infeasible network, the step is rejected. If the splitting results in an improved network, the step is accepted and the algorithm proceeds.



Exchangers A and B before stream splitting

The same exchangers after stream splitting

Figure 3: The topology of Stream Splitting

When a part of the stream is split, the flow is distributed between the two branches at a certain ratio called the splitting ratio. The load on the exchanger in each branch is a function of this ratio because it determines the flow rate. Ideally, it should be left to the model to decide the optimal ratio, but the side-effect would be that the model would become non-linear. This would be a problem for ITRI, as the constraint solvers in SICStus only optimise linear systems.

As an approximate solution, we use a heuristic to calculate the splitting ratio before optimising the network. By instantiating the ratio, the non-linearity is eliminated from the model. Our heuristic binds the splitting ratio to be equal to the MCP (flow rate) ratio on the other side of the exchangers. This means that, if the splitting is on the hot side, the splitting ratio becomes the ratio of the MCPs of the two cold streams containing these two exchangers and *vice versa*. This heuristic was recommended by our domain expert as a good approximation, since, thermodynamically, it minimises energy losses when the split streams are remixed. Though it does not guarantee to reach the optimal ratio, it significantly reduces the complexity of the problem.

Adding a new exchanger The addition of new exchangers is performed to create a heat load path between a hot and a cold utility with the view to further reducing energy consumption. (This is done by transferring load off the utilities onto the process exchanger.)

To create a new heat load path, the exchanger(s) to be added are feasible if the total savings in the utility cost will offset the implied capital expenditure within the payback period specified. It is worth mentioning that, in our system, it is assumed that a single exchanger is needed to create a heat load path. This assumption is made because otherwise there is a potential combinatorial explosion of solutions involving more and more exchangers. We could have relaxed this assumption by generating early failure in the search tree to exclude part of the search space, but this would need an *ad hoc*, case by case rule fix, which is not our goal here. We wish to test the feasibility of our system on broader application problems, rather than constraining it to a specific case.

When we add a new exchanger, its position must be determined. This is done by exhausting all possible positions to place the exchanger, which results in generating multiple solutions with different structures. Note that the maximum number of these positions is polynomial in the number of exchangers, so this approach is computationally tractable. The solution set might be filtered, as a possible extension of our work, by flexibility and safety constraints.

4.4 The mathematical model of the network

The main functions of our linear programming model are to handle the load shifting and to guarantee the feasibility of the network. The load shifting is handled by minimising the total cost of the utilities. The feasibility conditions of the network are represented by the set of constraints in the linear programming model.

The optimisation model does not include all types of costs in the objective function. In the ITRI model, the utility cost is the only cost included in the objective, which means that the model is linear, within the context of the isothermal mixing assumption, and so can be solved with a linear programming optimiser such as CLP($\mathcal{R}$) in SICStus Prolog 3 (which is our implementation language). The nonlinear part of the objective is then handled by the search mechanism, which evaluates all the outcomes from the linear programming model using the nonlinear part of the objective. Because of this approximation, the optimal solution of the problem may not be reached, as it may be that the optimal solution does not achieve the maximum heat recovery, but the algorithm will generate its solutions within the specified payback period. It is worth mentioning that even if the optimal solution is skipped, it can be retrieved easily by constraining a lower bound on the total utility loads. This option is available in our implementation.

The mathematical model incorporates two groups of constraints: those which stem from the first law of thermodynamics; and those deriving from the second law. The former are applied for each exchanger on each stream in the network; the latter are applied each time two exchangers are compared.

The first law of thermodynamics states that energy shall be conserved. This constrains the load on the hot side of an exchanger to be equal to that of the cold side minus losses to the atmosphere. Furthermore, hot streams are being cooled and cold streams are being heated, and this sets additional constraints on their inlet and outlet temperatures. Mathematically,

$$\begin{gathered} \mathrm{Inlet}_{\mathrm{hot}} \geq \mathrm{Outlet}_{\mathrm{hot}} \\ \mathrm{Inlet}_{\mathrm{cold}} \leq \mathrm{Outlet}_{\mathrm{cold}} \\ |\alpha \times \mathrm{MCP} \times (\mathrm{Inlet} - \mathrm{Outlet})| = \mathrm{Load} \\ \mathrm{Load}_{\mathrm{hot}} = \mathrm{Load}_{\mathrm{cold}} \end{gathered}$$

The second group of constraints embody the second law of thermodynamics. This involves the following constraints: the difference between the inlet temperature on the hot side and the outlet temperature on the cold side should be greater than or equal to the minimum approach temperature; the difference between the outlet temperature on the hot side and the inlet temperature on the cold side should be greater than or equal to the minimum approach temperature;

$$\begin{gathered} \mathrm{Inlet}_{\mathrm{hot}} \geq \mathrm{Outlet}_{\mathrm{cold}} + \Delta_t \\ \mathrm{Outlet}_{\mathrm{hot}} \geq \mathrm{Inlet}_{\mathrm{cold}} + \Delta_t \end{gathered}$$

The nonlinear equation for the area of an exchanger is calculated using the following equation:

$$\mathrm{Area} = \mathrm{Load}/(\frac{(h_h * h_c)}{(h_h + h_c)} * \frac{(\mathrm{Inlet}_{\mathrm{hot}} - \mathrm{Outlet}_{\mathrm{cold}}) - (\mathrm{Outlet}_{\mathrm{hot}} - \mathrm{Inlet}_{\mathrm{cold}})}{e^{\frac{\mathrm{Inlet}_{\mathrm{hot}} - \mathrm{Outlet}_{\mathrm{cold}}}{\mathrm{Outlet}_{\mathrm{hot}} - \mathrm{Inlet}_{\mathrm{cold}}}}})$$

where h_h and h_c are the heat transfer coefficients on the hot and cold sides respectively.

4.5 Assumptions and limitations of ITRI

In the current version of the system, there are some assumptions and limitations, mainly because of either the time limitations of the project or practical issues. These assumptions and limitations are discussed in the following points.

1. The HEN is assumed to be in a steady state. This assumption is generally considered reasonable when using mathematical programming methods.
2. Heat exchangers are assumed to support single streams. This assumption could be relaxed easily later on for any further extension of the algorithm.
3. The input information is assumed complete. This is generally considered a reasonable assumption in steady state models.
4. Inlet and Outlet temperatures are assumed to be fixed. As long as one is not interested in the interaction of the HEN design with the optimal design of flow-sheets, this is not a restrictive assumption.
5. The algorithm makes the isothermal assumption. This means that, in case of splitting, all the exchangers in the splitting set have the same inlet and outlet temperature. This is still reasonable as it is generally not economic to break the isothermal constraint because it is considered as a loss in energy.
6. A single exchanger is assumed to be enough to generate a heat load path. This is because of the search space. As was mentioned above, this could be relaxed easily by including problem specific knowledge to prune the search tree, or by setting a practical upper bound such as 2 or 3 on the number of exchangers to be considered for path creation.

5 Testing and evaluation

The system was tested on a number of different examples from the chemical engineering literature. The examples were chosen to guarantee the generality of the test, and to cover most of the problems that might be found in solving practical applications. In this section, we discuss a selected example for illustration purposes.

The data for this example is taken from Ciric and Floudas (1989). Figure 4 shows the initial structure of the network, *i.e.*, before retrofitting. In the figure, S1 and W1 represent hot steam and cold water respectively, MCP is the material specific coefficient, and h is the amount of energy loss. Table 1 shows the area of the existing exchangers and the set of matches.

Exchanger	**Area (m^2)**	**Original Match**
1	45.06	H2-C1
2	12.50	H1-C2
3	33.09	H3-C1
4	23.50	H1-C3
5	05.75	S1-C1
6	05.39	H1-W1
7	11.49	H2-W1

Table 1: The area of the existing exchangers

In Table 2, the cost data for the example is presented. The overall heat transfer coefficient for this problem was reported to be $0.8KWm^{-2}K^{-1}$ and the installation costs were all zero. The solution produced by our method represents a considerable improvement over the existing solution which had

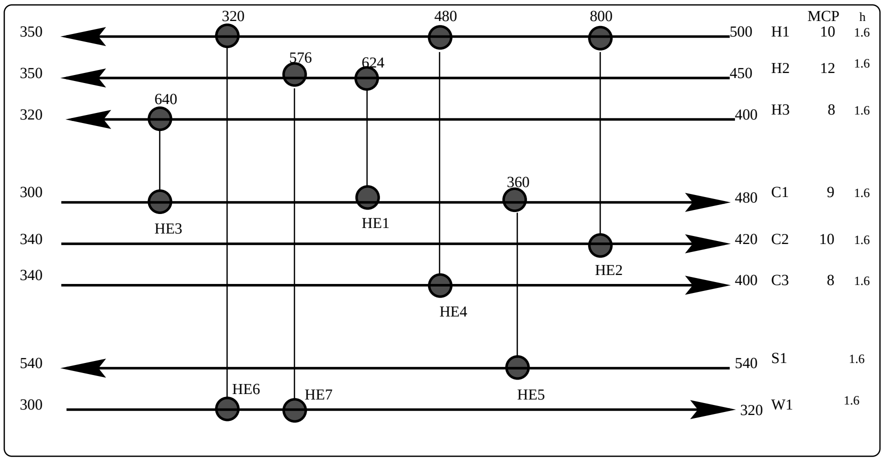


Figure 4: The existing HEN

cost category	**cost**
C_1	0
C_2	400
C_3	3460
C_4	171.4
C_5	80.00
C_6	20.00

Table 2: The costs data

claimed optimality. Our solution had an investment cost of \$9056 with a saving of \$35920 per year. This resulted in a payback period of 0.252 years. The total added area is 45.839m^2 and no new exchanger is needed. The best published solution to this problem (Yee and Grossmann 1987) reported an investment cost roughly twice this value, for the same energy savings, and the first solution, also via mathematical programming, claiming optimality, reported costs three times those reported by the proposed method, without improving the energy recovery. Furthermore, the solution presented here is considerably less complicated, involving no new exchangers in contrast to the earlier solutions which required a number of repipes and stream splits. The exchanger area before and after the retrofit is shown in Table 3.

The new configuration, generated by our program, and confirmed as correct by our domain expert, is shown in Figure 5. Further examples and analysis may be found in Abbass (1997).

6 Conclusion

In this paper, we have presented a novel algorithm for retrofitting an existing heat exchanger network, which uses traditional logic programming search, combined with linear constraint solution technology. It seems very efficient from a practical point of view, when compared with existing conventional solutions to the same problem.

Some standard examples from the literature were redesigned in a unified framework to facilitate

Exchanger	Area before the retrofit (m^2)	Area after the retrofit (m^2)
1	45.06	63.547
2	12.50	38.742
3	33.09	33.090
4	23.50	18.974
5	5.75	5.406
6	5.39	6.500
7	11.49	4.300

Table 3: Area of exchangers in example 1 after retrofitting

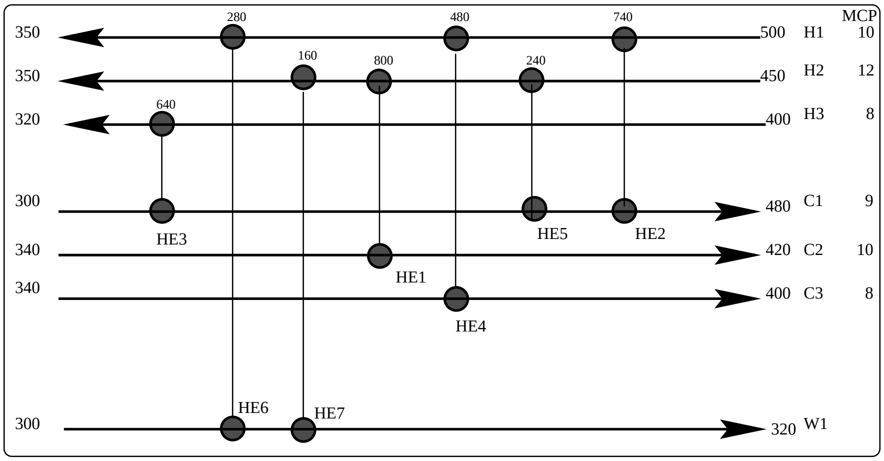


Figure 5: The retrofitted HEN of example 1

the comparison of different approaches for further research in the area.

We have used guidelines introduced by Lakshmanan and Bañares-Alcántara (1996) for retrofit by inspection as a basis for our new automated system. The guidelines were formalised, improved, and implemented as a constraint logic program. The resulting system outperforms the existing techniques from the chemical engineering literature in most problems, though there are some difficult examples which require solutions of non-linear equations.

Throughout this work, we have emphasised the idea of *integrating heuristic search with mathematical optimisation techniques* to improve the quality of the solutions resulting from the heuristic techniques as well as to improve the efficiency of applying the optimisation techniques. This approach seems not to have been considered before in the chemical process synthesis domain.

References

Abbass, H. A. (1997). An intelligent tool for process synthesis. Master's thesis, Department of Artificial Intelligence, University of Edinburgh.

Androulakis, I. and V. Venkatasubramanian (1991). A genetic algorithmic framework for process design and optimization. *Computers and Chemical Engineering 15*.

Asante, N. and X. Zhu (1997). An automated and interactive approach for heat exchanger network retrofit. *Chemical Engineering Research and Design 75*(A3), 349–360.

Briones, V. and A. Kokossis (1996). A new approach for the optimal retrofit of heat-exchanger networks. *Computers and Chemical Engineering 20*.

Ciric, A. and C. Floudas (1989). A retrofit approach for heat exchanger networks. *Computers and Chemical engineering 13*(6), 703–715.

Garrard, A. (1996). Mass exchange network synthesis using genetic algorithms. Master's thesis, Dept. of Chemical Eng., Edinburgh University.

Gundersen, T. and L. Naess (1988). Review paper: The synthesis of cost optimal heat exchanger networks, an industrial review of the state of the art. *Computers and Chemical Engineering 12*(6), 503–530.

Kelser, M. and R. Parker (1969). Optimal networks of heat exchange. *Chem. Eng. Prog. Symp. Ser. 92*(111).

Kobayashe, S., T. Umeda, and A. Ichikawa (1971). Synthesis of optimal heat exchange systems: An approach by the optimal assignment problem in linear programming. *Chem. Eng. Sci. 26*.

Lakshmanan, R. and R. Bañares-Alcántara (1996). A novel visualisation tool for heat exchanger network retrofit. *Industrial and Engineering Chemistry Research 35*, 4507–4522.

Lewin, D. (1998). A generalized method for HEN synthesis using stochastic optimization - ii. synthesis of cost optimal networks. *Computers and Chemical Engineering 22*(10), 1387–1405.

Lewin, D., H. Wang, and O. Shalev (1998). A generalized method for HEN synthesis using stochastic optimization - i. general framework and MER optimal synthesis. *Computers and Chemical Engineering 22*(10), 1503–1513.

Linnhoff, B. and E. Hindmarsh (1983). The pinch design method for heat exchanger networks. *Chemical Engineering Science.*

Nishida, N., G. Stephanopoulos, and A. W. Westerberg (1981). A review of process synthesis. *AIChE J. 27*(3), 321–351.

Shokoya, C. (1992). *Retrofit of Heat Exchanger Networks for debottlenecking and energy savings.* Ph. D. thesis, Department of Chemical Engineering, UMIST, Manchester, UK.

Tjoe, T. and B. Linnhoff (1986, April). Using pinch technology for process retrofit. *Chemical Engineering.*

Umeda, T. et al. (1979). A thermodynamic approach to the synthesis of heat integration systems in chemical processes. *Computers and Chemical Engineering 3*, 273–282.

Viswanathan, M. and L. B. Evans (1987). Studies in the heat integration of chemical process plants. *AIChE J. 33*(11), 1781–1790.

Wang, H., O. Shalev, and D. Lewin (1997). A generalised method for hen synthesis using stochastic optimisation: (i) general framework and mer optimal synthesis. *Computers and Chemical Engineering.*

Yee, T. et al. (1990). Simultaneous optimisation models for heat integration-I area and energy targeting and modeling of multi-stream exchagers. *Computers and Chemical Engineering 14*(9), 1151–1164.

Yee, T. and I. Grossmann (1987). Optimization model for structural modifications in the retrofit of heat exchanger networks. Technical report, Engineering Design Research Center.

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

Yee, T. and I. Grossmann (1991). A screening and optimisation approach for the retrofit of heat-exchanger networks. *Industrial Engineering Chemical Research 30*(1), 146–162.

Yee, T., I. Grossmann, and Z. Kravanja (1990). Simultaneous optimisation models for heat integration-III process and heat exchanger network optimisation. *Computers and Chemical Engineering 14*(11), 1185–1200.

A Mathematical Problem Formulation

As the literature does not contain a formal, mathematical definition of the HEN retrofit problem, we present the formulation that was used in this research. It is considerably more general than the majority of mathematical programming formulations and includes the facility to declare individual stream match and repipe costs.

Given:

- A set of hot streams $HS = \{H_1(T_{in_{H_1}}, T_{out_{H_1}}, HMCP_1, Elist(H_1)), H_2(T_{in_{H_2}}, T_{out_{H_2}}, HMCP_2, Elist(H_2)), \ldots, H_n(T_{in_{H_n}} T_{out_{H_n}}, HMCP_n, Elist(H_n))\}$ where $T_{in_{H_i}}$ and $T_{out_{H_i}}$ represent the inlet and outlet temperatures for hot stream H_i respectively, $HMCP_i$ represents the mass flow specific heat product of the ith hot stream, and $Elist(H_i)$ is a list with all sets of exchangers on this stream. Each element in this list takes the form (exchanger-label, predecessors-list, successors-list, inlet-temperature, outlet-temperature, load, area, MCp)
- A symmetric list of cold streams, $CS = \{C_1(T_{in_{C_1}}, T_{out_{C_1}}, CMCP_1, Elist(C_1)), C_2(T_{in_{C_2}}, T_{out_{C_2}}, CMCP_2, Elist(C_2)), \ldots, C_n(T_{in_{C_n}}, T_{out_{C_m}}, CMCP_m, Elist(C_m))\}$.
- A set of hot utility streams $SS = \{S_1(T_{ins_1}, T_{outs_1}), S_2(T_{ins_2}, T_{outs_2}), \ldots, S_x(T_{ins_x}, T_{outs_x})\}$ where T_{ins_i} and T_{outs_i} represent the inlet and outlet temperatures of stream S_i respectively.
- A set of cold utility streams $WS = \{W_1(T_{in_{W_1}}, T_{out_{W_1}}), W_2(T_{in_{W_2}}, T_{out_{W_2}}), \ldots, W_y(T_{in_{W_y}}, T_{out_{W_y}})\}$.
- A set of existing heat exchangers (matches) $M_{ex} = \{M_1(Type, Area, Alpha), \ldots, M_k(Type, Area, Alpha)\}$ where *Type*, is the type of match that the exchanger provides (*he* for process-process match, *hu*, for process-hot-utility match, and *cu* for process-cold-utility match), *Area* is the area of the exchanger, and *Alpha* is the fraction of energy lost to the ambient from the exchanger.
- A set of costs $K = \{K_1, K_2, K_3, K_4, K_5, K_6\}$ where K_1 is the cost of moving a heat exchanger, K_2 is the cost of a single repipe, K_3 is the cost of purchasing a new heat exchanger, K_4 is the cost of additional square meter of area, K_5 is the cost per KW/year for hot utilities, and K_6 is the cost per KW/year for cold utilities. Note that these costs are not restricted to single numbers, but could be vectors or even matrices depending on the problem.
- Define $HLstream = \sum_{i=1}^{n}(T_{in_{H_i}} - T_{out_{H_i}}) * HMCP_i$ to be the total load on the hot streams, $CLstream = \sum_{j=1}^{m}(T_{in_{C_j}} - T_{out_{C_j}}) \times CMCP_j$ to be the total load on the cold streams, $Lmatch = \sum_{l=1}^{k} \Omega_k \times Load_k$ (where Ω_k is 1 if the kth exchanger matches a hot stream with a cold stream and 0 if the exchanger matches a process stream with a utility) to be the total energy recovered by the match set, $Hutility = CLstream - Lmatch$, to be the total hot utility needed, and $Cutility = SLstream - Lmatch$ to be the total cold utility needed.

- Define

$$\mathsf{Payback} = \frac{\mathsf{initial\,cost\,of\,total\,utilities} - \mathsf{final\,cost\,of\,total\,utilities}}{\mathsf{moving\,cost} + \mathsf{re\_piping\,cost} + \mathsf{area\,cost}}$$

The retrofit problem now consists of finding a new set of matches and exchanger/utility loads that satisfy the driving force and material balances constraints with an acceptable trade-off between the minimisation of the total cost comprised of the set K, and the maximisation of the improved energy recovery (represented by *Lmatch*) to satisfy a user specified payback period.

In this definition, the total cost is comprised of the six types of costs that have been given in the definition. K_1 takes effect if the order of the exchanger in the new match set has been changed without altering the pair of streams being matched by it. K_2 takes effect if either of the streams matched has been changed. K_3 takes effect for all new exchangers (p). K_4 takes effect if the area of the exchanger in the new match set has been increased. K_5 & K_6 are simply multiplied by *Hutility* and *Cutility*, (calculated as shown in the definition) respectively to get the total cost of the utilities.

It is assumed, at the current time, that the temperatures of the hot utilities are high enough to be matched with any cold stream in any temperature interval, and symmetrically that the cold utilities are cold enough to be matched with any of the hot streams. However, there is nothing fundamental to the method that requires this, and this restriction should be lifted in the future.