Integrating optimisation and search: An Intelligent Tool for Chemical Process Synthesis

Content Areas: Search; Problem solving
Tracking Number: 108

Abstract

Conventional optimisation techniques suffer from an inability to reason about their decisions, as well as from the complexity of their search space. Heuristics can help then to reach a solution quickly, though it may not be optimal. Integration between optimisation techniques and heuristics can enrich the problem solving process by providing partially optimal solutions in a reasonable time with partial explanation. We present a novel method for solving an important problem in chemical process synthesis: the retrofit of heat exchanger networks. Our method combines expert intuition, encoded as a set of heuristics, with linear programming, an optimization technique, using the CLP paradigm. More interestingly, the integration facilitated the inclusion of domain knowledge which could not be represented in optimisation. Our results are significantly superior to those reported in the chemical process synthesis literature, particularly those obtained from existing pure optimisation techniques.

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 show 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 optimization techniques, which ignore domain specific knowledge, themselves.

Process Synthesis [7] is 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 aspects: reaction of chemicals, separation of the results, and *utilities* (sources of external heat or cooling). In traditional process design, the reaction stage defines the tasks for the subsequent design of the separators. Heating and cooling requirements for the reaction and separation stage are determined last, being regarded as auxiliary operations in the process. However, this naïve approach can increase the total operational cost.

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 design of *Heat Exchanger Networks* (HENs) has been receiving a great deal of attention.

Heat exchanger networks utilise the energy content of streams of hot fluid to heat cold ones. Broadly speaking, the hot streams flow through shells and the cold streams flow in tubes within the shells. 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 require heating. Sometimes, it is necessary to supplement the function of a HEN with external 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 [4] give a comprehensive review of the field. In a typical grass-roots problem, the aim is to find the 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. 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 guarantees that the temperature of a hot stream would always be 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 only 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. For example, we might often want to avoid removing an exchanger physically because this represents a loss in capital cost, but a medium term cost benefit analysis might actually confirm that this is indeed the best course of action.

So the retrofit designer is concerned not only with the constraints involved in a grass-roots problem, but also with legacy constraints as well. The existing exchangers represent explicit constraints on the problem as it is uneconomic and ecologically unfriendly to get rid of them; they 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 use of exhaustive search without an effective way to prune the search space. We propose such a method here.

The solution of the grass-roots HEN problem is addressed in the literature by means of two popular approaches. In the first, the problem is solved by *pinch technology* [6]. 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-root 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 [11; 12].

The harder problem of HEN retrofit is solved using a combination of pinch targeting and optimisation models [13; 2]. This combination aims to overcome the burden of computation in optimisation models by developing some initial boundaries using the pinch method.

Retrofit by inspection was suggested by Tjoe and Linnhoff [9] and further implemented by Lakshmanan and Ba‘ nares-Alcántara [5]. 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.

The rest of this paper is structured as follows. In Section 2, previous attempts for applying AI techniques to solve the problem are presented followed by a formal representation of the problem and the proposed method in Section 3. In Section 4, a case-study is presented, and conclusions are drawn in Section 5.

2 AI approaches to HEN design

Knowledge based systems have been introduced in HEN design to overcome the difficulties with pure mathematical models, and sometimes to reduce the complexity of the initial problems to be solved by mathematical programming methods. A number of Artificial Intelligence (AI) techniques have been applied to grass-rooting, but until now, none has been tried on the retrofit problem.

Androulakis *et al.* [1] present a *genetic algorithm* (GA) for HEN design. The chromosome is defined as a fixed-size collection of matches between hot and cold exchanger streams. Each gene within the chromosome has an allele set comprising an alphabet of $n_h \times n_c$ characters, which denote all possible matches between n_h hot streams and n_c cold streams. However, this is not an ideal representation, as, in a real synthesis problem, the number of exchangers is not known in advance. So one really needs a variable-length chromosome. Androulakis *et al.* argued that a variable length chromosome would result in an untractable search space. However, we suggest that a variable-length chromosome *with a length bounded above* could be adopted because one can simply predict such an upper bound from the initial stages of HEN design.

Also Androulakis *et al.* argue that, unless the fitness value is a monotonic function of the string size, no conclusions can be drawn when two strings of different sizes are compared. This disregards the fact that the fitness function can be adapted to make the comparison effective: one could penalise the fitness of each string with its length, or with its total cost.

Garrard [3] used a GA on a mass exchange network, with promising results.

Wang *et al.*[10] propose an updated version of Androulakis' GA. They use multiple levels, where the chromosome is represented by a string of dimension $n_c \times n_l$, where n_c is the number of cold streams and n_l is the number of levels. The alleles are numbers from 0 to n_h, where n_h is the number of hot streams. If the gene takes the value 0, this means that there is no exchanger present. Once a structure has been defined, a lower level LP is solved to maximise the heat loads of the exchangers. The modified algorithm also allows stream splitting (see below). This unfortunately results in a *non-linear programming* (NLP) problem in evaluating fitness of each chromosome. So although the method succeeds in generating optimal structures, it is impractical for realistic problems, because of the expensive computations required.

3 Automating HEN Retrofit

Our system, called ITRI (Intelligent Tool for Retrofit by Inspection), has three main components The first is a domain-specific representation. The second is a set of heuristic rules, elicited from literature surveys and from a domain expert, which guide the search. The third component applies network optimisation – mainly for generating the initial approximate mathematical model.

3.1 Problem representation

This formal definition forms a general framework for standardising the HEN retrofit problem; it is useful in itself, as there are no existing unified case studies for comparison. To go with the definition, we introduce a standardised set of data needed for HEN retrofit. In presenting our solutions (below), we have standardised the data needed for our examples. This has required some (safe) assumptions for parameters which are omitted in the literature.

The HEN retrofit problem can be formally stated thus:

Given

- A set of hot streams $HS = \{H_1(T_{I_{H_1}}, T_{O_{H_1}}, HMCP_1, Elist(H_1)), ..., H_n(T_{I_{H_n}} T_{O_{H_n}}, HMCP_n, Elist(H_n))\}$ where $T_{I_{H_i}}$ and $T_{O_{H_i}}$ represent the inlet and outlet temperatures for hot stream H_i respectively, $HMCP_i$ represents the material coefficient specific of the ith hot stream, and $Elist(H_i)$ is a sequence of all sets of exchangers on hot stream i; each element takes the form ⟨Exchanger-label, predecessors-list, successors-list, inlet-temperature, outlet-temperature, load, area, mcp⟩.
- A set of cold streams $CS = \{C_1(T_{I_{C_1}}, T_{O_{C_1}}, CMCP_1, Elist(C_1)), ..., C_n(T_{I_{C_n}}, T_{O_{C_m}}, CMCP_m, Elist(C_m))\}$ where $T_{I_{C_j}}$ and $T_{O_{C_j}}$ represent the inlet and outlet temperatures for cold stream C_j respectively, $CMCP_j$ represents the material coefficient specific of C_j, and $Elist(C_j)$ is a sequence of all sets of exchangers on cold stream i. The representation is as for hot streams, above.
- A set of hot utilities $SS = \{S_1(T_{I_{S_1}}, T_{O_{S_1}}), ..., S_x(T_{I_{S_x}}, T_{O_{S_x}})\}$ where $T_{I_{S_i}}$ and $T_{O_{S_i}}$ represent the inlet and outlet temperatures of stream S_i respectively.
- A set of cold utilities $WS = \{W_1(T_{I_{W_1}}, T_{O_{W_1}}), ..., W_y(T_{I_{W_y}}, T_{O_{W_y}})\}$ where $T_{I_{W_i}}$ and $T_{O_{W_i}}$ represent the inlet and outlet temperatures of stream W_i respectively.
- A set of existing heat exchangers $M_{ex} = \{M_1(Type, Area, Alpha), ..., M_k\ (Type, Area, Alpha)\}$ where *Type*, is the type of match that the exchanger matches (*he* for stream-stream match, *hu*, for stream-hot utility match, and *cu* for stream-cold utility match), ***Area*** is the area of the exchanger, and ***Alpha*** is the amount of energy loss corresponding to the material of the exchanger.
- A set of costs $C = \{C_1, C_2, C_3, C_4, C_5, C_6\}$ where C_1 is the cost of moving a heat exchanger, C_2 is the cost of re-piping one stream, C_3 is the cost of purchasing a new heat exchanger, C_4 is the cost of additional square meter of area, C_5 is the cost per KW/year for hot utilities, and C_6 is the cost per KW/year for cold utilities.

Define

- $HLstream = \sum_{i=1}^{n}(T_{I_{H_i}} - T_{O_{H_i}}) * HMCP_i$ to be the total load on the hot streams, $CLstream = \sum_{j=1}^{m}(T_{I_{C_j}} - T_{O_{C_j}}) \times CMCP_j$ 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 k's exchanger matches a hot stream with a cold stream and 0 if the exchanger matches either a hot stream with a water or a cold stream with a steam) 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.
- $Payback = \Delta Utility\ cost \div Retrofit\ cost$ where $\Delta Utility\ cost = initial\ cost\ of\ total\ utilities - final\ cost\ of\ total\ utilities$ and $Retrofit\ cost = moving\ cost + repiping\ cost + area\ cost$

The problem of heat exchanger retrofit is to find a new set of matches, represented by new sets of hot and cold streams, that satisfies the driving force and material balance constraints with a reasonable trade-off between the minimisation of the total cost, $\sum_{i=1}^{6} C_i$, and the maximisation of the heat recovery represented by $Lmatch$ within a specific payback period.

It is assumed that the temperatures of the hot utilities are high enough to be matched with any cold stream in any temperature interval, and that the temperature of the cold utilities are small enough to be matched with any hot stream in any temperature interval.

3.2 The heuristic search mechanism

ITRI solves the HEN retrofit problem by applying four group of rules recursively while optimising the loads on the utilities. To control the combinatorial explosion thus produced, some heuristics were applied, as follows.

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. This involves both linear and non-linear costs, the linear cost being used during optimization, 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; and attempt is made to optimise the network using the linear part of the objective; then the total total objective (*i.e.* linear and nonlinear) is evaluated. If the solution results in a payback period more than three times the payback period required by the user, the algorithm rejects it and backtracks. The factor three was suggested by our domain expert as be a reasonable limit; we need it because of the non-linear part of the objective, to prevent ITRI being fooled by horizon effects.

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 the 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. If we reject the repiping step, the overall effect 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. That is, we restrict the search space without restricting the solution space.

Adding Exchangers In our 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 many cases.

Occurrence Checks In addition to the heuristics that are used to reason about the search space, a number of occurrence checks are 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 the unloaded exchanger.

The second check is on the payback of the current solution. If the current payback is less than the best payback found sofar, 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 latest solution.

The third check forbids loops within the heat path.

Finally, if the model is multi-optimal (according to the CLP(R) library used in the implementation), the algorithm creates a choice point and replaces the bindings of the variables with their infima and suprema.

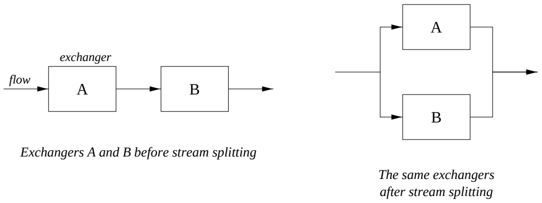
The Four Main Rule Groups

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, is performed at two points: before doing any change in the network; and after each change to the network. Loads are shifted by the linear programming model, which guarantees that the maximum load to be shifted in the network as a whole, and accordingly the maximum heat recovery, will be achieved. The model does this by minimising the total utility cost, which results in the minimum total utility loads. This has the disadvantage that it may skip an optimal solution if the optimality is not at the maximum heat recovery.

Exchanger reallocation 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*, where the heat exchange is not as efficient as it should be. This can be corrected by swapping the two exchangers. After doing so, 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 backtracks to find another two exchangers to be reversed. The algorithm continues the reallocation until no other useful interchanging in the network exists. By symmetry, only the exchangers on the hot side need to be tested. Because the exchangers are checked pairwise, the complexity of the check has an upper bound of the square of the number of exchanges – 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 1. An exchanger creates a bottleneck if it prevents a load from being shifted on a load path. This case is 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 to place the 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 1: 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 of this would be that the model would become non-linear. This would be a problem for ITRI, as its constraint solvers 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. Though it does not guarantee to reach the optimal ratio, it significantly reduces the complexity of the problem.

Adding a new exchanger There is a distinction between adding an exchanger to remove a bottleneck, as above, and adding one to create a heat load path. In the latter case, a new exchanger is added when the HEN has a cold utility and a hot utility with no heat load path between them.

To remove a bottleneck, it is more economic to split, if possible, rather than adding a new exchanger. To create a new heat load path, the exchanger(s) to be added are feasible if the total savings in the utility cost after the change will be greater than the capital cost needed for installing the new components. In our system, it is assumed that a single exchanger is needed to create a heat load path; 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, 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.

3.3 The mathematical model

The main functions of our linear programming model are handling the load shifting, and guaranteeing the feasibility of the network. 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.

In the ITRI model, the utility cost is the only cost included in the objective, which means that the model is linear, and so can be solved with a linear constraint solver such as CLP($\mathcal{R}$) in SICStus Prolog 3.5 (which is our implementation language). The nonlinear part of the objective is handled by Prolog's search mechanism.

The mathematical model incorporates two kinds of constraints: those covering the first law of thermodynamics; and those covering the second law. The former apply to each exchanger on each stream in the network; the latter apply each time two exchangers are matched.

There are four main constraints in the first group. The first two constrain the inlet and the outlet temperatures of each exchanger to be non-negative. The third relates the amount of energy loss to the material of the exchanger, the material coefficient, and the inlet and outlet temperatures with the load on the exchanger. The fourth constrains the inlet temperature to be greater than the outlet temperature for hot streams and *vice versa* for cold. For example, if the exchanger is on the hot side, the four constraints encoding the first law are:

$$\begin{gathered}\mathsf{Inlet} \geq 0 \\ \mathsf{Outlet} \geq 0 \\ \mathsf{Alpha} \times \mathsf{MCP} \times (\mathsf{Inlet} - \mathsf{Outlet}) = \mathsf{Load} \\ \mathsf{Inlet} \geq \mathsf{Outlet}\end{gathered}$$

If the exchanger is the first on the stream, its inlet temperature must equal the inlet temperature of the stream. Similarly, if it is last on the stream, its outlet temperature must equal that of the stream.

The second group of constraints embody the second law of thermodynamics: each exchanger obeys the law of conservation of energy. This involves three 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; and the load of the exchanger on the hot side must equal its load on the cold side.

4 Testing and evaluation

ITRI was tested on several 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 Tjoe and Linnhoff [8]. Figure 2 shows the structure of the network before retrofit. Table 1 shows the area of the existing exchangers and the set of matches.

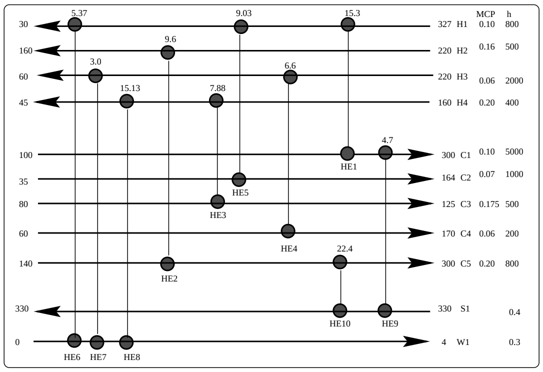


Figure 2: The existing HEN for the example

Exchanger	Area	Original Match
1	0.0003	H1-C1
2	0.0012	H2-C5
3	0.0009	H4-C3
4	0.0007	H3-C4
5	0.0008	H1-C2
6	0.3520	H1-W1
7	0.1237	H3-W1
8	0.6711	H4-W1
9	0.2357	S1-C1
10	0.7777	S1-C5

Table 1: Area of existing exchangers in example

In Figure 3, the network after retrofitting is shown and the new area compared with the original one is shown in Table 2. The total added area is 0.014904469m^2 with four new exchangers added to the current network. The total modification cost is \$17,840 and the total savings is found to be $18,405. This resulted in a 0.97 year payback period. The amount of energy recovered from this retrofit is 18.405kW. This example shows a great improvement over the results obtained by Tjoe. His final retrofit was with total utilities load 43.37kW and payback 1.8 years, whereas in our solution, we achieved total utilities load 32.195kW with only 0.97 year as a payback period. In terms of payback period, our solution was 46% better.

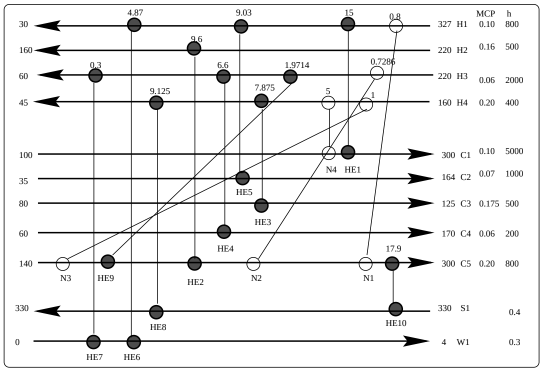


Figure 3: The HEN retrofit for the example

Exchanger	Area before the retrofit (m^2)	Area after the retrofit (m^2)
1	0.0003	0.001145000
2	0.0012	0.003130000
3	0.0009	0.004748000
4	0.0007	0.007260000
5	0.0008	0.001138000
6	0.3520	0.331400000
7	0.1237	0.016530000
8	0.6711	0.478900000
9	0.2357	0.000085360
10	0.7777	0.691405024
N1	0	0.000017469
N2	0	0.000149000
N3	0	0.000250000
N4	0	0.000967000

Table 2: Area of exchangers in example after retrofit

Figure 2 shows the initial structure of the network. Table 1 shows the area of the existing exchangers and the set of matches.

5 Conclusion

We have presented a novel AI algorithm for retrofitting an existing heat exchanger network, which uses tradition 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 the comparison of different approaches for further research.

We have used guidelines introduced by Lakshmanan and Bañares-Alcántara [5] for retrofit by inspection as a basis for our new automated system. They were formalised, improved, and implemented as a constraint logic program. Our system out-performs existing techniques from the chemical engineering literature by up to 50% 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 resulted 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

[1] I.P. Androulakis and V. Venkatasubramanian. A genetic algorithmic framework for process design and optimization. *Computers and Chemical Engineering*, 15, 1991.

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

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

[4] T. Gundersen and L. Naess. 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, 1988.

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

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

[7] C.G. Shokoya. *Retrofit of Heat Exchanger Networks for debottlenecking and energy savings.* PhD thesis, Department of Chemical Engineering, MSIT, 1992.

[8] T.N. Tjoe. *Retrofit of Heat Exchanger Networks.* PhD thesis, University of Manchester, 1986.

[9] T.N. Tjoe and B. Linnhoff. Using pinch technology for process retrofit. *Chemical Engineering*, April 1986.

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

[11] T.F. Yee et al. 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, 1990.

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

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