Using Qualitative, Numerical and Heuristic Knowledge to support Innovative Design

Nikitas M. Sgouros
Qualitative Reasoning Group
The Institute for the Learning Sciences
Northwestern University
1890 Maple Ave., Evanston IL 60201
E-mail: nikitas@ils.nwu.edu

Abstract

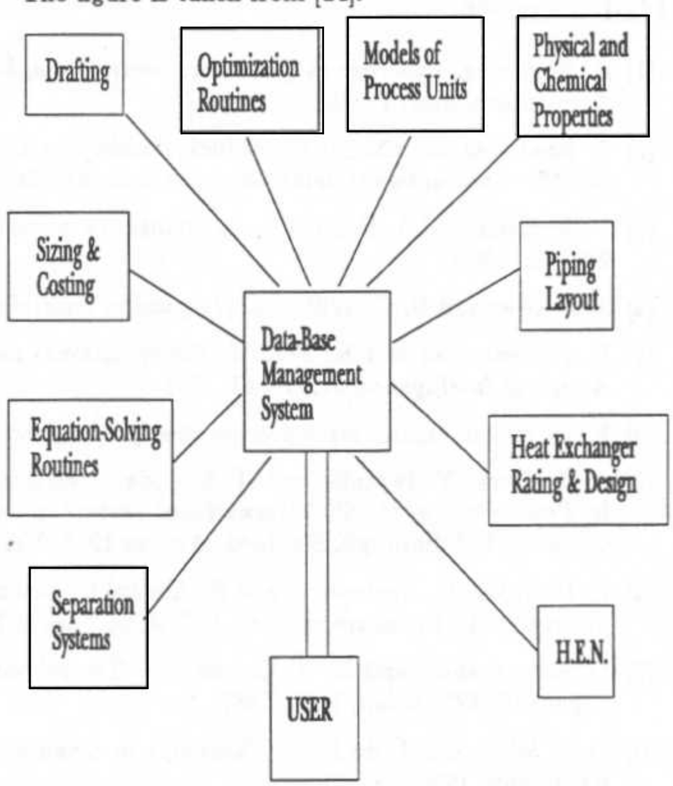
A computational account of innovative design that integrates qualitative, numerical and heuristic knowledge is presented. Qualitative and numerical models are used to represent the physical knowledge of the application domain. The design knowledge for the domain is encapsulated in a set of heuristic rules organized in terms of design strategies and a set of configuration synthesis rules that provide the procedures for changing the design description. The application area we are investigating is the synthesis of multicomponent separation sequences in chemical engineering. A system is presented that supports multiple evolutionary strategies for the design of separation systems. We show how such an integrated approach generates complex designs similar to those presented in the chemical engineering design literature.

Introduction

Design problems can be classified as belonging to one of three possible categories [13]. In a *routine design* problem, the components of design are known and there is a specific method for assembling these components. In *innovative design* the components of the design are known, but there is no straightforward method for assembling the components in a way that satisfies the specifications of the design problem. Finally, in *creative design* problems, not even the components of the design are completely known. The synthesis of separation sequences in chemical engineering is an example of an innovative design problem [12].

The dominant approach for solving design problems in most engineering fields today is the computer-aided design (CAD) methodology. The major operating assumption behind all the modern CAD packages is that the computer will provide a set of highly specialized support tools to the user in order to help him/her with the current problem [14]. Figure 1 shows an example of an advanced CAD environment for chemical process design. In this program a database management sys-

Figure 1: Typical computer-aided design environment. The figure is taken from [14].



tem is used to coordinate the information flow between a set of specialized subroutines that deal with various design subtasks such as equipment sizing, physical and chemical properties, numerical optimization routines, etc. While this approach makes it possible to derive numerical solutions for complex descriptions of artifacts, it relies exclusively on the human designer to formulate the design alternatives and coordinate the use of all the programs during the process. In particular, such an environment does not possess either explicit models of the design knowledge (e.g. heuristics, design strategies) that are needed to drive the process or representations of physical knowledge that are able to automatically create and analyze design alternatives at the desired level of detail.

This paper proposes a computational account of innovative design that tries to overcome the limitations of CAD systems. This approach provides explicit representation formalisms for the physical and design knowledge that are used in design along with a method that allows their integration in the process. Qualitative reasoning is used to provide a formal modeling language for representing the qualitative aspects of the physical knowledge that is applied in the design task.

In the next section we briefly describe the physical principles and the major design methods for the synthesis of multicomponent separation sequences in chemical engineering. Due to the wide use of separation processes in almost every chemical plant, the design of separation systems is one of the most important areas of research in process synthesis. Sections 3 and 4 present the architecture and the algorithm used in the design system respectively. Experimental results for our implementation are given in Section 5. In Section 6 our design approach is compared with current research in the AI and Chemical Engineering communities. The final section is a conclusion.

The synthesis of separation sequences

The synthesis of separation sequences problem can be defined as follows [9]: given a feed stream of known conditions (i.e., composition, flow rate, temperature, pressure), synthesize a process that can isolate the desired (specified) products from the feed at minimum cost. Table 1 provides an example of a separation design problem.

The separation processes we are interested in are ordinary and extractive distillation for multicomponent mixtures. Distillation is one of the most widely used processes in chemical engineering [8]. It involves the separation of the components of a mixture based upon differences in their tendencies to evaporate at a given temperature. In a binary (two-component) mixture the component with the highest tendency to evaporate is called the *volatile* component of the mixture. The other component is called the *non-volatile* component of the mixture. Distillation results in two products:

Feed		
Component	**Component Name**	**Mole Fraction**
1	n-Hexane	.3333
2	Benzene	.3333
3	Cyclohexane	.3334
Desired Products		**Conditions**
Product	**Component**	$T = 37.8°C$
1	1	$P = 1.033\ kg/cm^2$
2	2	Total Flow Rate =
3	3	= 170.1 kg mol/h

Table 1: Problem definition for the C6 separation. In this case we have a mixture of three components (n-Hexane, Benzene and Cyclohexane) which we want to separate into pure component products. The composition, the flow rate and the temperature and pressure of the input mixture are given.

the product at the top of the column is called the distillate, while the one at the bottom of the column is called the bottom product. The net effect of the process is an increase in the concentration of the volatile component in the distillate and of the non-volatile component in the bottom product. The analysis used to describe distillation columns that accept multicomponent mixtures as feed is similar to the analysis used for binary distillation. In these cases two of the components of the mixture with neighboring boiling points are selected as the *key components* of the separation. Usually, the one with the lower boiling point is called the *light key* while the other one is called the *heavy key*. All the other components are called *nonkeys*. Under this description, distillation causes most of the light key and all the more volatile nonkeys to appear in the distillate product, while most of the heavy key and all the less volatile nonkeys end up in the bottoms. Extractive distillation differs from its ordinary counterpart in that it involves the addition of a new component to a mixture in order to facilitate the separation of the system by distillation [8]. It usually involves two distillation columns. The first one is used to accomplish the desired separation while the second one is used to recycle the added component.

The design methods for the synthesis of separation sequences can be divided into three categories [10]: heuristic, evolutionary and algorithmic methods. This research concentrates on evolutionary methods. Heuristic methods can be subsumed under the evolutionary paradigm while algorithmic methods rely explicitly on well-developed numerical methods that produce optimal design solutions, but are computationally inefficient and cumbersome to use in the majority of cases [10].

Evolutionary methods consist of three basic steps [11]: (i) The creation of an initial flowsheet using a number of heuristic rules. (ii) The application of a set of evolutionary rules to the current flowsheet. A design strategy is used to optimize the application of these rules. The purpose of this step is to make systematic and small changes to the current flowsheet in the hope of creating a better design. (iii) A comparison between all the flowsheets that were generated in the previous step in order to select the optimal one.

Steps (ii) and (iii) consitute a loop that is executed until there are no more changes suggested by the evolutionary rules or until the changes proposed by these rules result in suboptimal designs.

Over the years a number of evolutionary strategies have been proposed [10]. All of these approaches use overlapping subsets of 19 major design heuristics for synthesizing the initial separation sequence together with a set of evolutionary rules that differ between methods.

The Design System

The design system consists of five components which belong to one of two categories. The first category includes the components that describe the physical knowledge that we have about the domain. These include:

(1) A domain theory that represents in qualitative terms the physical knowledge for the separation processes we are dealing with.

(2) A set of equations that corresponds to the domain theory given in (1).

The second category includes the components that represent our design knowledge for the given domain. These include:

(1) A set of design heuristics.

(2) A set of design strategies that optimize the application of heuristic knowledge.

(3) A set of configuration synthesis rules that implement the design decisions of the system by modifying the current design description.

The input to the system consists of:

(1) A set of initial values and descriptions for some of the parameters of the separation system. These may include the names and specifications of the desired products along with the composition of the initial feed to the system.

(2) The modeling assumptions that we are willing to make in the design.

(3) The names of the design strategies we will try to use in the system.

The system generates as output a list of designs that consist of a qualitative description for the proposed separation sequence along with numerical values for the separation parameters.

Currently, the implementation of the system is based on an assumption-based truth maintenance system (ATMS) [3]. A rule engine (ATMoSphere) [4] provides the interface to the ATMS. All of the strategies, the heuristics and the configuration synthesis rules are translated into ATMoSphere rules.

The following sections describe each component in more detail.

```
(defView
 (Unobtained-Column-Products ?desired-products ?column)
 Individuals
 ((?column
   :Type Distillation-Column
   :Conditions
   (Examine ?column)
;; The feed to the column is a multicomponent mixture (M C S).
   (Column-Feed (M-C-S ?components ?phase ?stage) ?column))
;; The actual products for the column based on the separation
;; scheme in the Preconditions field of the view.
  (?products
   :Conditions
   (Column-Products ?products (?l-k ?h-k) ?column))
  (?env
   :Conditions
  (Value-of (A (Atmospheric-Pressure ?env)) ?atm-pres ?a-eqn))
  (?l-k :Type Substance ;; The light key for the separation.
      :Test (member ?l-k ?components))
  (?h-k :Type Substance ;; The heavy key for the separation.
       :Test (and (member ?h-k ?components)
                  (not (eql ?h-k ?l-k))
;; Decide on whether the proposed keys have neighboring
;; boiling points in order for the separation to be a
;; feasible one.
                  (neighboring-boiling-points?
                   ?h-k ?l-k ?atm-pres ?components)))
;; The desired products that were specified in the design
;; specifications for the original problem.
  (?desired-products
   :Test (and (subsetp ?desired-products ?products)
              (not (subsetp ?products ?desired-products)))
   :Conditions (Desired-Products ?desired-products ?column)))
 Preconditions
 ((Consider
   (Possible (Separation distillation (?l-k ?h-k) :In ?column))))
 Relations
 ((Missing-Products
   ?desired-products ?products (?l-k ?h-k) ?column)))
```

Figure 2: Typical model fragment in the domain theory. This particular view computes the products from the design specifications that were not recovered using the current separation scheme.

Representing Physical Knowledge

Central to all the reasoning styles in science or engineering is the creation of representations that capture the understanding that engineers or scientists have about physical phenomena. These representations comprise the *physical knowledge* for a domain. Physical knowledge is organized around *models*, i.e. structured descriptions of the phenomena of interest. Each model contains:

- Sets of modeling assumptions under which the physical knowledge for a domain is valid.
- Sets of relevant features for the phenomena of interest. These include the introduction of quantities that measure important attributes of the system we are examining (e.g. quantities that measure structural features or cost in the case of design) along with methods for calculating them.
- The constraint relations between the parameters of the system.

Our design approach uses a combination of qualitative and numerical models to capture all these aspects of models. Qualitative formalisms are used to construct and describe design alternatives in ways that are consistent with the modeling assumptions made by the designer. Numerical models are used to analyze these alternatives based on their qualitative descriptions.

The Qualitative Domain Theory The qualitative models that we use contain high-level descriptions of ordinary and extractive distillation columns for multicomponent mixtures. The language we use to represent our domain theory is based on Qualitative Process Theory (QPT) [6]. As [2] notes, QPT is especially suitable for modeling chemical processes, because its process-centered ontology is able to capture the physical principles on which unit operations in chemical engineering are based. Figure 2 provides an example of a model fragment for this domain. These models are able to:

(1) Use the physical principles that describe separation processes in conjunction with our modeling assumptions in order to present the design system with all the separation alternatives. For example, the design system uses the qualitative domain models to compute all the possible keys for a given column based on the ordinal relations between the boiling points of the feed components. The whole method is explicitly predicated on a sharp separation approximation for multicomponent columns.

(2) Provide a natural way for analyzing the current design description at the desired level of detail. In particular, the system uses a set of heuristics for deciding on a particular separation for each column in the sequence. The results of this heuristic analysis match with the preconditions that control the activation of some of the model fragments in the domain theory. As a result, detailed qualitative and numerical models are activated only for the separation selected from the heuristics. This improves the efficiency of the analysis phase of design.

In addition, qualitative models can provide more focus during the analysis phase of design through the use of control predicates that direct the attention of the system on specific parts in the design description. The *(Examine ?column)* statement in the Individuals field of Figure 1 is an example of such a predicate. The particular model fragment is instantiated only for the columns specified as the ones to be examined in the design description.

The equations The set of equations used are either approximations for some of the separation parameters, calculations of physical properties for the various substances that are contained in the process feed or linear equations relating the input and output parameters with the required specifications for each column. Each equation is attached to a model fragment in the domain theory, and becomes active when that model fragment is implied from the design decisions made by

```
(defHeuristic Heuristic-2
 :Class Separation-Method-Selection
;; The predicates in the Conditions slot are the antecedents
;; of an ATMoSphere rule.
 :Conditions
((Possible
  (Separation
   extractive-distillation ?keys1 :In ?column)) :Var ?f1
 (Possible (Separation distillation ?keys2 :In ?column)) :Var ?f2)
;; The action slot is the body of the ATMoSphere rule.
 :Action ((prefer ?f2 :Over ?f1 :Justified-by (?f1 ?f2))))

IF there is an extractive distillation alternative
      for the current column
  AND there is an ordinary distillation alternative
      for the current column
THEN prefer the ordinary distillation alternative.
```

Figure 3: Typical heuristic form and its interpretation.

the system. This organization allows the instantiation of numerical models that are consistent with the level of detail and the modeling assumptions used by the system.

Representing Design Knowledge

Heuristics Heuristics are organized into classes according to the physical principles they apply or the general design heuristics they instantiate. Figure 3 provides an example of the form used to represent heuristic rules in the system. The current heuristic library consists of 36 rules organized in terms of 21 classes. They cover 13 out of the 19 design heuristics from the process synthesis literature used for synthesizing the initial separation sequence along with nine evolutionary rules for refining the initial structure.

Strategies Design strategies are plans for sequencing the execution of the various classes of heuristic rules in ways that were found capable of producing optimal designs. These strategies are explicitly represented in our approach. This scheme enables the design system to reason about their appropriateness for a given task. In addition, it allows the system to be used as a testbed for experimenting with different strategies that can solve a specific problem. Figure 4 provides an example of the form used to represent a design strategy in the system. Each strategy representation includes the heuristic classes that it uses and the preconditions under which it should be applied. In addition, it contains control information that optimizes the application of heuristic knowledge. This information is stored in the *ATMS-Context*, *Focus-Predicates* and *Execution-Order* slots.

The *ATMS-Context* slot determines the problem-solving context under which the heuristic rules are going to be applied. There are two possible problem-solving contexts. Under the *:Implied-By* context specified in Figure 4, all the application conditions for each heuristic rule must be implied by the current focus environment in the ATMS in order for the heuristic rule

```
(defStrategy Nath-&-Motard-Evolutionary-Strategy
;; The names of the heuristic classes that are used.
:Heuristic-Classes (Avoid-Distillation Delay-MSA-Removal)
;; The bindings for the variables in the Conditions slot are the
;; ones used in the rest of the form. The predicates in the
;; Conditions slot become the antecedents of an ATMoSphere rule.
:Conditions
((Separation-System ?system)
 (Consider (Design-Complete ?system))
 (Consider
  (Evolutionary-Strategy-for ?system Nath-&-Motard)))
:ATMS-Context :Implied-By
;; The instantiated predicates in this slot will be appended to
;; the beginning of every heuristic rule that belongs to this strategy.
:Focus-Predicates ((Apply-Strategy-to ?system))
;; The body of the ATMoSphere rule in which the defStrategy form
;; is translated consists of the contents of the Action slot
;; along with a set of functions in which the Execution Order slot
;; is translated.
:Action ((assume (Apply-Strategy-to ?system))
:Execution-Order
(:SERIAL Avoid-Distillation Delay-MSA-Removal))

IF we have completed the design of a separation system
   AND we were told to use the Nath-&-Motard strategy
       to evolve the current design
THEN mark the separation system we are going to evolve (the
       :Action part) and then apply the heuristics that are
       stored under the class Avoid-Distillation followed by
       the heuristics stored under the class
       Delay-MSA-Removal (the :Execution-Order slot).
```

Figure 4: Typical strategy form and its interpretation. MSA in the form stands for mass separating agent and it refers to the new component that is added to a mixture in the extractive distillation case. This particular form represents part of the evolutionary strategy for the Nath & Motard design strategy.

to trigger [5]. In our case, this focus environment contains all the major design decisions made by the system before the execution of the strategy. In the *:In* context all the applications conditions for each heuristic rule must be believed for the rule to fire.

The *Focus-Predicates* slot contains a set of predicates that will focus the application of heuristic rules to certain columns or systems in the separation sequence.

Finally, the *Execution-Order* slot determines the mode (serial, parallel or any combination of them) under which the heuristic classes are going to be applied.

Currently, the system is able to support two of the most significant evolutionary strategies for the synthesis of separation sequences [9], [11].

Configuration Synthesis Rules This is a set of rules that provide a set of procedures for changing the current design description. For example, one such rule will fire whenever the products for a column in the sequence have been determined. It will then create descriptions for the columns that are going to be connected to the product streams (distillate and/or bottom products) of the current column that do not correspond to any of the desired products in the problem specification. Currently the system contains 31 such rules.

The Design Algorithm

The algorithm used by the system is an iterative cycle that consists of the following steps:

(1) Analyze qualitatively the current design description. The qualitative analysis performed by the system is the same with the one used in SIMGEN [7] and it is used to determine the operating conditions under which each model fragment in the domain theory can be active.

(2) Construct the focus environment that provides the basis for the activation of the numerical models, the heuristics and the configuration synthesis rules in each design cycle. This is an ATMS focus environment that consists of the major design decisions made using the heuristics. For example, whenever the heuristic analysis decides on a particular separation for a column, it updates the current focus environment to reflect this decision.

(3) Solve the numerical equations that are implied by the current focus.

(4) Apply the design strategies and the configuration synthesis rules that are implied by the current focus. Most of the time this step results in the generation, selection and scheduling of design alternatives along with the updating of the focus environment.

The last two steps constitute an inner loop which is applied until the heuristics suggest no more changes to the current focus environment. This scheme provides the system with a way of controlling the level of detail in which each design alternative is analyzed. For example, the system has to decide for every column generated by the configuration synthesis rules which of several possible separations is the optimal. When this decision process is done, the current focus environment is updated to include only the proposed separation. This in turn activates the numerical models that describe the detailed features for the proposed separation only.

Two conditions must hold for the design cycle to terminate: (i) The configuration synthesis rules suggest no modifications to the current design description. (ii) There are no more design alternatives to explore.

Examples

The Search Space At each step in separation system design the main problem is to decide on the kind of separation process that is going to take place in the current column. We tested our system in examples where the choice was between ordinary or extractive distillation processes. For each kind of separation process that is examined there is a list of alternatives depending on the key components that are going to be used. Therefore, in the case of a mixture of N components to be separated into N pure component products using M separation methods, the number of possible separations is given by [15]:

$$R = \frac{[2(N-1)]!}{N!(N-1)!} M^{N-1} \quad (1)$$

Results We have tested the system on two popular examples from the process synthesis research literature: the C6 separation problem that involves the separation of a 3-component mixture of n-Hexane, Benzene and Cyclohexane into its individual components and the n-butylene purification system involving the separation of a 6-component mixture into subsets of its components [9]. Both strategies were used in each one of these examples resulting in a total of 4 design problems. We will describe the behavior of the system in the first example using the Nath & Motard design strategy [9] and we will give only performance results for the rest of the design cases.

Table 1 describes the details of the C6 separation problem. The system accepts the problem description and uses the configuration synthesis rules to generate the first column in the sequence. The qualitative models compute two possible separations for the first column based on the boiling points of the feed substances: (i) the one in which the light key is the n-Hexane and the heavy key is the Benzene and (ii) the one in which the light key is the Benzene and the heavy key is the Cyclohexane. A number of model fragments that describe high-level features for the candidate separations become active in the domain theory. These in turn activate a set of equations that calculate high level parameters for the design alternatives (e.g. the separation factors for all the candidates at the reference conditions). The system now tries to apply its design strategies. The set of heuristics for computing the initial flowsheet in the Nath & Motard strategy is activated. The heuristic analysis uses the results of the qualitative analysis and the values for the parameters of the proposed separations to prune the design space. The second separation is rejected as too difficult. The system assumes that the first alternative is viable and proceeds to instantiate more detailed qualitative and numerical models for this separation (e.g. the Fenske-Underwood-Gilliland method for calculating the separation parameters). The configuration synthesis rules complete the description of the column and based on the products of the proposed separation instantiate the next column in the sequence. This cycle is repeated until a sequence for recovering all the desired products has been proposed and its cost has been established. Figure 4a depicts the initial sequence.

At this point the conditions for the activation of the evolutionary part of the design strategy have been established. These rules suggest an extractive distillation unit with n-Hexane as the light key and Cyclohexane as the heavy key for the first column in the sequence. A new design cycle starts that adopts this change and the flowsheet in Figure 4b is proposed. It is found to be cheaper than the one in Figure 4a. Once again the evolutionary rules are activated and they attempt to modify the design in Figure 4b. The second distillation column in this sequence is replaced by a distillation unit that isolates the Cyclohexane as the distillate product. Another design cycle is generated that results in the sequence shown in Figure 4c. This design is found to be cheaper than the previous one. The evolutionary rules do not suggest any further modifications and the design process terminates. The design results are consistent with the ones reported in [9].

Figure 5: Proposed flowsheets for the C6 separation problem. The columns marked with an asterisk (*) are extractive distillation units. Every other column in the sequence is an ordinary distillation unit. Furfural is a polar solvent used in the extractive distillation case.

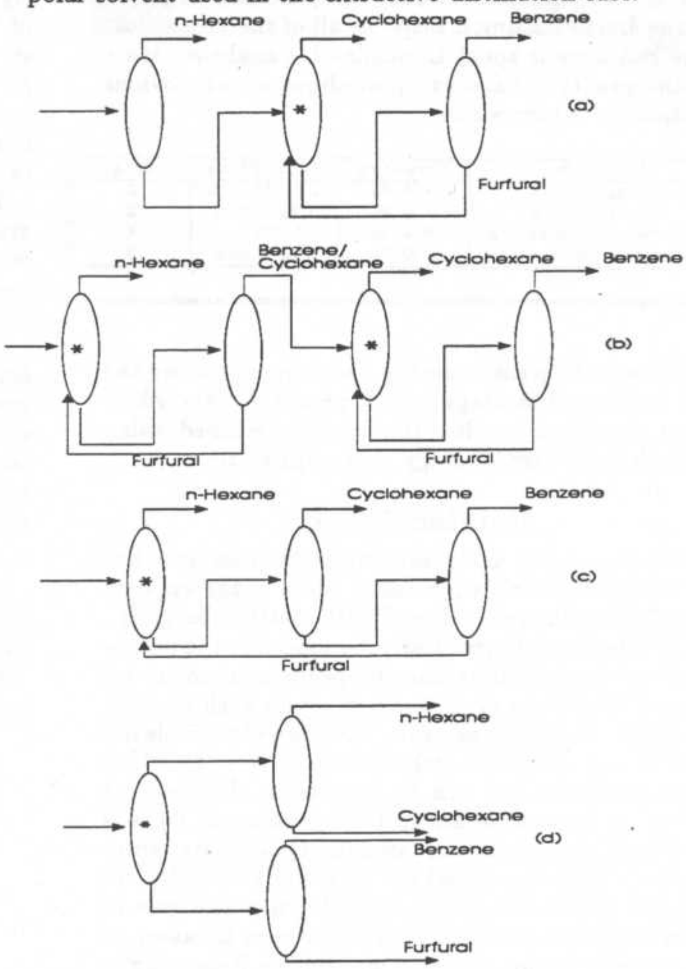


Table 2 presents the performance results for the 4 design problems. Both methods proposed the same design for the n-butylene purification problem. We know of no published results for the application of the Seader & Westerberg method on the C6 separation problem. The final design suggested by the system using this

Table 2: Performance results for the 4 design problems. The Designs column in the table refers to the number of designs examined by the system until an optimal solution was found. The system was run on an IBM RS/6000, Model 530, with 128 MB of RAM running Lucid Common Lisp. In all of the cases most of the run time is spent in qualitative analysis. More efficient qualitative analysis procedures are an obvious direction for future work.

Problem	Strategy	Run Time	Designs
C6 separation	N & M	1 hr 35 min	3
C6 separation	S & W	1 hr 9 min	2
n-Butylene purification	N & M	> 10 hr	4
n-Butylene purification	S & W	4 hr 6 min	3

method was different from the one proposed using the Nath & Motard strategy (see Figure 4d). According to our numerical results, the solution reached using the Nath & Motard strategy is the optimal.

Related Work

Qualitative models and reasoning techniques have not been widely used in design tasks. Most of the applications in this field are concerned with creative design [1], [16]. While these efforts clearly demonstrate the potential of using qualitative knowledge for supporting innovation, they have problems coming up with schemes that could efficiently constrain the space of possible designs. In addition, they lack rigorous criteria by which the proposed designs can be evaluated. We decided to focus on innovative design for two reasons: (i) It is widely used in engineering tasks (ii) Qualitative representations that can model the physical knowledge involved in innovative design will play a major role in automating this process. This is the case because, as this research demonstrates, Qualtitative Reasoning is able to provide formal languages for representing the qualitative aspects of the knowledge used by engineers in design.

We view our research as more closely related to the design research efforts in the process engineering community [14], [2]. Most of the members of this community are interested in developing computational accounts that allow the automatic construction of models for chemical processes at multiple levels of detail. Like them, we are convinced that modeling languages that capture the assumptions and approximations used in describing physical systems along with the qualitative and numerical knowledge that is necessary to understand their behavior, will provide the basis for supporting complex engineering activities such as design, diagnosis or control. We believe that the development of such formalisms will not be restricted to the process engineering domain but will scale up to cover a large part of engineering knowledge.

Conclusion

We have described a computational account of innovative design that integrated the physical knowledge represented as qualitative and numerical models with the design knowledge represented in heuristic rules and design strategies. By combining all these different types of knowledge the system can support multiple design strategies. This research was applied to the synthesis of separation sequences for multicomponent mixtures in chemical engineering. Preliminary results indicated that this approach generates designs similar to the ones found in the process synthesis literature.

Evolutionary design methods are not limited to separation systems but are general methods for solving process synthesis problems [10]. We believe that our approach can support more types of design problems in chemical engineering.

Directions for future work include improving the efficiency of the qualitative analysis component in the system; applying this approach to other engineering domains; creating mechanisms for generating explanations for the various design decisions; and developing a set of criteria for choosing among different design strategies.

Acknowledgements

Thanks to Ken Forbus and Wesley Burghardt (of the ChemE Dept. at Northwestern) for useful comments. This research has been supported by the NASA Johnson Space Center.

References

[1] Aelion, V., Cagan J., Powers, G. J., Inducing Optimally Directed Innovative Designs from Chemical Engineering First Principles, *Computers & Chemical Engineering*, vol. 15, no. 9, 1991.

[2] Catino, C. A., Grantham, S. D., Ungar, L. H., Automatic generation of qualitative models of chemical process units, *Computers chem. Engng.* 15, 583-599, 1991.

[3] deKleer, J., An assumption-based truth maintenance system, *Artificial Intelligence*, 28, 1986.

[4] Forbus, K. D., The qualitative process engine, in *Readings in Qualitative Reasoning about Physical Systems* edited by Weld, D. S. and deKleer. J., Morgan Kaufmann Inc, 1990.

[5] Forbus, K. D., deKleer, J., Focusing the ATMS, *AAAI-88*, August 1988.

[6] Forbus, K. D., Qualitative Process Theory. *Artificial Intelligence*, 24, 1984.

[7] Forbus, K. D. and Falkenhainer, B., Self-Explanatory Simulations: Scaling up to large models. *AAAI-92*, July 1992.

[8] King, C. J., Separation Processes, McGraw-Hill, Inc., 1971.

[9] Nath R., Motard, R. L., Evolutionary Synthesis of Separation Processes, *AIChE Journal*, vol. 27, no. 4, July 1981.

[10] Nishida N., Stephanopoulos G., Westerberg, A. W., A Review of Process Synthesis, *AIChE Journal*, vol. 27, no. 3, May 1981.

[11] Seader, J. D., Westerberg, A. W., A Combined Heuristic and Evolutionary Strategy for Synthesis of Simple Separation Sequences, *AIChE Journal*, vol. 23, no. 6, November 1977.

[12] Siletti, C. A., Design of Protein Purification Processes by Heuristic Search, in *Artificial Intelligence in Process Engineering* edited by Mavrovouniotis, M. L., Academic Press Inc., 1990.

[13] Sriram, D., DESTINY: A Model for Integrated Structural Design, *Journal for AI and Engineering*, October 1986.

[14] Stephanopoulos G., Artificial Intelligence in Process Engineering-Current State and Future Trends, *Computers & Chemical Engineering*, Vol. 14, No. 11, 1990.

[15] Thompson, R. W., King, C. J., Systematic Synthesis of Separation Schemes, *AIChE Journal*, 18, 941, 1972.

[16] Williams, B., Invention from First Principles via Topologies of Interaction, PhD Thesis, MIT, June 1989.