A REPRESENTATION FOR ALGORITHMIC EXPANSION OF ENGINEERING DESIGNS

Vital Aelion, Jonathan Cagan, Gary J. Powers

EDRC 24-90-92

A Representation for Algorithmic Expansion of Engineering Designs

by

Vital Aelion
Department of Chemical Engineering

Jonathan Cagan*
Department of Mechanical Engineering

Gary J. Powers
Department of Chemical Engineering

Carnegie Mellon University
Pittsburgh, PA 15213

* Author to whom correspondence should be addressed

Abstract

A knowledge representation based on first principles and a library of techniques for expanding the space of design alternatives are reviewed. They have been developed for the algorithmic innovation of engineering designs.

The knowledge representation consists of an algebraic design objective, a set of algebraic equality and inequality constraints, and variable and constraint typing specifications. The algebraic relations model the fundamental physical and chemical phenomena, as well as manufacturing, safety, and performance considerations. Within this representation, variable typing describes physical, processing and manufacturing characteristics of variables within a design. Constraint typing describes the region(s) in which each constraint is applicable in a design. This representation may be useful in applications which are amenable to mathematical representations, and require additional information on variable characteristics and constraint applicability, to achieve their reasoning goals.

The expansion techniques introduce new independently modeled features in a known design. Presently there are two expansion techniques: *Dimensional Variable Expansion*, DVE, and *Input Variable Expansion*, IVE. These techniques are the means for modifying design topology. DVE and IVE have been defined as mathematical operators, to manipulate our knowledge representation. The concept of design space expansion is useful independently of knowledge representation.

The knowledge representation and the library of design space expansion techniques have been combined into the 1stPRINCE design methodology for innovation in engineering design. This methodology uses optimization information to decide which expansion technique may produce improved designs and induction to examine limiting behavior. Starting from an initial design, 1stPRINCE has algorithmically innovated interesting engineering concepts, such as the *electric bus*, the *tapered* and *hollow beam*, the *wheel*, and the *plug flow reactor*.

1. Introduction

Design may be described as a search for a good solution in a space of possible configurations that satisfy the user's goals. Design frequently involves three important aspects:

1. a set of design objectives,
2. a representation of design alternatives, and
3. a search technique.

In this paper we review (a) a knowledge representation which is based on first principles, (b) a library of design space expansion techniques for innovating designs, and (c) a search technique which is based on optimization.

The first principle knowledge of this paper is represented as an algebraic objective, a set of algebraic engineering equations, a design topology, and a set of specifications of constraint and variable types. The equations are derived from detailed physical, chemical, manufacturing, safety, and performance considerations. This knowledge representation is useful in domains which can be described mathematically, and require additional information on variable characteristics and constraint applicability, to achieve their reasoning goals.

The expansion techniques are mathematical heuristics which innovate designs by introducing new features into a known design. Optimization information is used to decide which expansion technique is applicable.

The first principle knowledge representation, the library of expansion techniques, and the optimization-based search have been incorporated into the 1stPRINCE design methodology for innovation in engineering design. The knowledge representation and the expansion techniques may be useful to other research efforts independently.

This paper unifies various pieces of this research effort, which have been published in [Cagan and Agogino, 1987, 1991a, and 1991b] and in [Aelion, *et al.*, 1991 and 1992]. After a discussion of each of these research issues, the 1stPRINCE methodology is applied to the design of a vehicle fueling strategy. Other applications are also summarized.

2. Classes of Engineering Design

We categorize design into three main classes based on the resulting artifact: *routine*, *innovative*, and *creative* designs [Cagan and Agogino, 1991a]. We differentiate between the artifact and process in our definitions, and characterize processes based on the type of artifacts they typically produce.

A *routine design* has a fixed set of features. A routine design process involves finding the best solution among known alternative design topologies. The challenge in routine design is to develop design objectives that adequately characterize the user's goals and search techniques capable of finding the best design in a reasonable computational time.

In an *innovative design*, new features are introduced to known designs. A technique which introduces these features to a design class is an innovative design process. While a well-defined objective and a powerful search technique are important in innovative design, the emphasis is in finding ways to expand the space of design alternatives.

Finally, *creative designs* are not based on known designs. In this category of designs the emphasis is placed almost exclusively on finding design solutions in previously unexplored design spaces. Creative designs could result from the development of new technologies, or the application of technologies different from the ones already in use.

The library of expansion techniques, presented in section 5, is capable of introducing new features into known designs, and supports innovative engineering design. Other work on algorithmic innovation is reviewed in the following section.

3. Related Literature on Innovation in Engineering Design

Traditionally research in artificial intelligence has produced designs by decomposing and manipulating existing design fragments [McDermott, 1977; Roylance, 1980; Mitchell, *et al.*, 1983; Ressler, 1984; Mitchell, *et al.*, 1985; Brown and Chandrasekaran, 1985; Mittal, *et al.*, 1986; Tong and Sriram, 1991]. Decomposition expedites search; however, a large class of engineering designs (*e.g.*, structures, mechanisms) cannot be easily decomposed, especially when new features are desired. Design reasoning systems capable of *innovative* designs usually are not based on the decomposition of the problem.

Innovation often occurs by expanding the space of engineering alternatives to introduce new design features. Other researchers have developed theories for design innovation. Murthy and Addanki [1987] developed the PROMPT system which uses heuristics based on first principles to reason about mechanical structures. PROMPT represents the various design models of engineering equations in a graph, called a *graph of models*, and uses domain-specific heuristics to modify the models. The approach looks for constraint satisfaction and uses numerical approaches to analyze and modify the structures.

Lenat [1982, 1983a, 1983b] developed the EURISKO system which also uses heuristics to mutate a frame-based LISP representation of a design. These mutations create new design configurations which may possess innovative characteristics. The system mutates LISP structures that are considered most interesting based on the number of occurrences of a LISP phrase and the success of designs with this phrase. In EURISKO the heuristics themselves are subjected to mutation, which leads to new heuristics. EURISKO has been most successful in designs with large and unexplored spaces.

Dyer, *et al.*, [1986] developed the EDISON project, a framework for invention by heuristic-based mutation and analogy. EDISON uses a fixed knowledge base that represents

the decomposition of a device to derive new combinations or uses of devices. A pre-defined set of mutations on given devices is used to achieve sub-goals and create new devices.

Maher, *et al.*, [1989] also examined the use of heuristic-based analogy and mutation of prototypes to derive new structural configurations. They developed a frame-based representation, called a *prototype*, that models function, structure, and the relationship between function and structure, of a device. By searching through a prototype for specified slots, their methodology can determine whether the prototype satisfies design requirements, partially satisfies the requirements, or cannot satisfy the requirements at all. The prototype then can be used unchanged, or mutated by domain-specific heuristics to satisfy problem requirements.

Ulrich and Seering [1988] pointed out that decoupled approaches to design have limited applications in mechanical design and examined the use of *function sharing* to derive new designs of dynamic systems. They deleted certain features from fluid devices and developed a hierarchy of hindrances which may result from the modification, features that cause the hindrances, and modifications to remove the hindrances from the system. They modeled the actual device in a decomposition of form, but maintained coupling through a sufficient model of the physics of the problem.

Joskowicz and Addanki [1988] related function to geometry via a space of kinematic configurations to derive innovative kinematic pairs. A complete mathematical mapping between geometry and kinematics is required to reason about the coupling between form and function. By examining the bounds between feasible and infeasible solutions, new devices can be derived.

Faltings [1991] employed qualitative reasoning to map relationships between function and form for the conceptual design of kinematic pairs. He also derived configuration spaces to aid in the conceptualization of an object, and derived a qualitative model of the configuration feasibility rather than the precise mathematical formulation of Joskowicz and Addanki. Faltings used a first-order predicate logic representation of geometry and inference to search within the representation.

Mitchell [1989] reasoned about shapes and their grammars in creative design. Mitchell looked at issues of emergence, ambiguity and discontinuity in shape interpretation, instability in the grammatical rules for carrying out shape computations, and non-monotonicity in reasoning about shapes. Shapes are represented as a collection of maximal geometric lines and shape grammars describe a language for modifying the model [Stiny, 1980].

Williams [1991] used abstract first principle representations of dynamic systems to derive topological configurations. The interaction-based design approach focused on coupling between components to solve configuration problems. A topological map between form and function for a general domain and one for a specific device instance were built from classes of

physical structures. The mapping modeled fundamental constitutive laws, such as continuity of fluid flow. Solutions with minimal sets of interactions were derived.

4. First Principle Knowledge Representation

First principle knowledge representations model problems based directly on the fundamental physical and chemical phenomena, as well as on manufacturing, safety, and performance considerations. There are many choices of first principle knowledge representations. A detailed representation, for example, may involve partial differential equation models of the system. Less detailed representations may be based on order of magnitude information, for example [Mavrovouniotis and Stephanopoulos, 1988]. A more abstract choice is a list of pairs of interacting variables, including the sign of the interaction (first partial derivative information), for example [Williams, 1991]. The more detailed the representation, the higher the precision in modeling a phenomenon, and the higher the complexity in analyzing a design.

Our representation uses algebraic relations in symbolic or in numerical form. Designs are described by an *algebraic objective* and a set of *algebraic constraints*. The representation also models fundamental characteristics of each quantity and each relation in the system, by *typing* the variables and constraints. Designs can have multiple *regions*. A region is a section of a design, which may be independently modeled and may have independent properties [Cagan and Agogino, 1991b]. The connectivity among regions is described by a *topology*. All of this information is provided by a designer.

The first principle information is captured by the notion of a *primitive-prototype*, based on an objective function, $f(\mathbf{x})$, to be minimized over a set of variables, $\mathbf{x}$, bounded by a set of equality and inequality constraints, $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ and $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$:

$$\begin{aligned} &\textit{Minimize} && f(\mathbf{x}) \\ &\textit{Subject to} && \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & && \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

We define five variable types. The first two, namely the *system* and *region* variables, designate whether these variables depend on the physical size of the system or a have a regional effect. The next three variables are subsets of these variable types and have been used by 1stPRINCE to expand the design space.

System variables, x^s, express quantities that depend on a characteristic size of a system. Examples of such variables include weight, reactor volume, capacitor charge and mass flowrate. System variables are replicated when new regions appear in the design. Note that

system variables have an extensive quality, as defined in the field of thermodynamics [Sandler, 1977].

Region variables, x^r, express quantities that describe a property of a design region. Such quantities are independent of all the region characteristic sizes. Region variables include temperature, pressure, density, Young's modulus and thermal conductivity. Region variables are replicated when new regions appear in the design. Region variables have an intensive quality, as defined in thermodynamics.

Assignment variables, x^a, express quantities that have contributions from multiple design regions, or that appear in specific design locations, such as the first and the last region. These variables facilitate the accounting within a design. Examples that account for contributions from multiple regions include total weight of a design, average density, total sales and total cost. Examples of variables which are specific to a particular design region include entering and exiting chemical reactant concentration.

In addition to the variable types defined above, we define two more variable types, which are subclasses of the types already defined. *Dimensional variables* are a special subclass of system variables that denotes the physical dimensions of a design, such as length, radius, or volume. *Input variables* are another subclass of system variables that describes *quantities to be processed* by a design. Examples of input variables include material flows in chemical reactor designs and loads in structural designs.

Constraints are classified in four types: *serial, parallel, boundary*, and *assignment constraints*. *Serial constraints*, c^s, are functions only of assignment variables and apply across all design regions. *Parallel constraints*, c^p, apply to individual design regions. *Boundary constraints*, c^b, are defined for each boundary between neighboring design regions. Finally, *assignment constraints*, c^a, are the definitions of the corresponding assignment variables.

Based on these definitions, the primitive-prototype becomes:

$$\textit{primitive-prototype} = f \text{ bounded by } \{c^s\} \cup \{c^p\} \cup \{c^b\} \cup \{c^a\} \text{ over } \{x^s\} \cup \{x^r\} \cup \{x^a\}.$$

Within this representation, searching for the best design involves minimizing the objective, while satisfying the design constraints. The space of solutions is limited by a fixed set of features. Design space expansion is one way to remove this limitation.

5. Design Space Expansion: A Method for Innovation in Engineering Design

Design space expansion introduces new features into a known design. This is achieved by creating new design regions, each of which can be independently modeled and assigned independent properties (independent intensive variables). All expansion techniques have a

similar basic algorithm, presented in Figure 1.

In this algorithm, step 1 is the identification of a region for expansion. In steps 2 and 3, a number of replicate regions is chosen and a new design topology is created. The connectivity of this new topology depends on which particular expansion technique is applied. In step 4, assignment constraints update the symbolic or the numerical values of assignment variables for the new topology. Step 5 accounts for the possible creation of new boundary constraints for each new region boundary. Step 6 is the creation of parallel constraints for each region in the new topology. Finally, in step 7, the objective function and the serial constraints are updated, based on the new values of the assignment variables.

BEGIN (*Any Expansion*)
1. **Identify a region for expansion.**
2. **Choose a number of replicate regions (default = 2).**
3. **Create new design topology.**
4. **Apply assignment constraints.**
5. **Create new boundary constraints.**
6. **For each region create parallel constraints.**
7. **Update objective function and serial constraints.**

END

Figure 1. General expansion algorithm

Two expansion techniques are currently available: *Dimensional Variable Expansion,* DVE, and *Input Variable Expansion,* IVE. DVE, shown in Figure 2, focuses on a region with a dimensional variable, in this case Dim 1. The initial design, which comprises of a single region in this case, is expanded into multiple regions along Dim 1. Each new region is independendy modeled and assigned new properties.

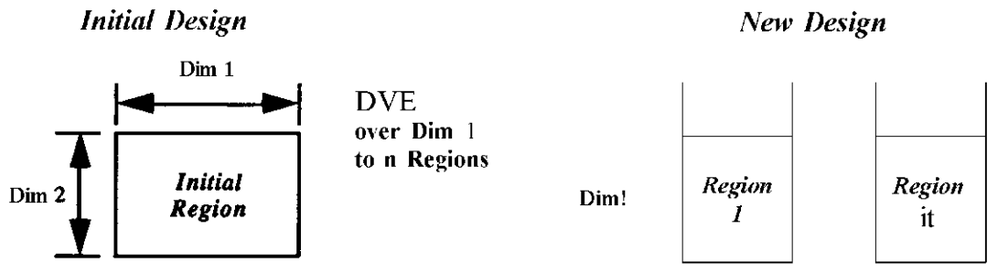

Figure 2. Conceptual illustration of DVE

An example application of DVE involving a beam is shown in Figure 3. The weight of a beam under flexural load is to be minimized, subject to a yield stress constraint. Application of DVE over the length, a dimensional variable, produces the expanded design space of a

stepped beam. Application of D VE along the radius, another dimensional variable, would produce a beam with several concentric regions.

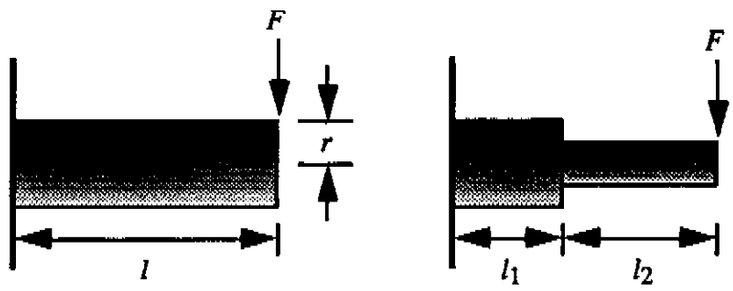

Figure 3. Application of DVE on a beam example

IVE, on the other hand, proposes the *parallel processing* of an input into design regions whose topologies are replicates of the initial design region, as shown in Figure 4. In doing so, IVE automatically introduces new features in the design, which may lead to better designs. In Figure 4, Input 1 is distributed to the multiple design regions which appear in the new design. The newly created design regions can be independently modeled. Application of IVE to the beam design over the load, an input variable, proposes designs of multiple beams, each of which carries a fraction of the original load, as shown in Figure 5.

DVE and IVE initiate a library of formal expansion techniques. These techniques are the means of expanding the space of possible design solutions, and provide the power to innovate designs. They have been developed for the knowledge representation presented in section 4, but their principle may be extended to other representations.

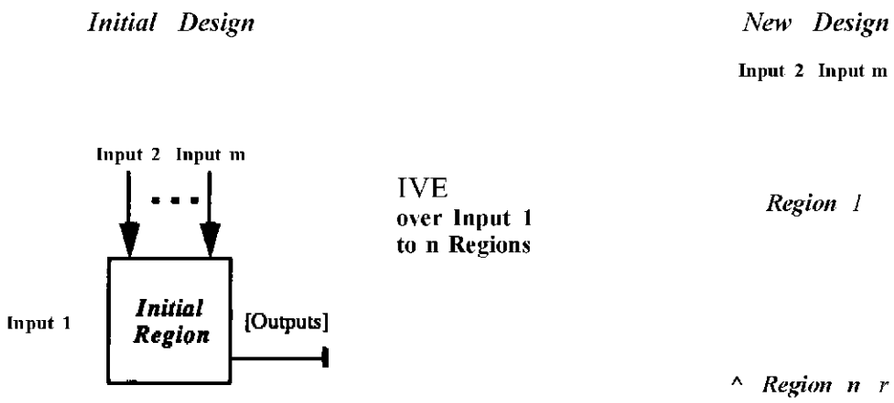

Figure 4. Conceptual illustration of IVE

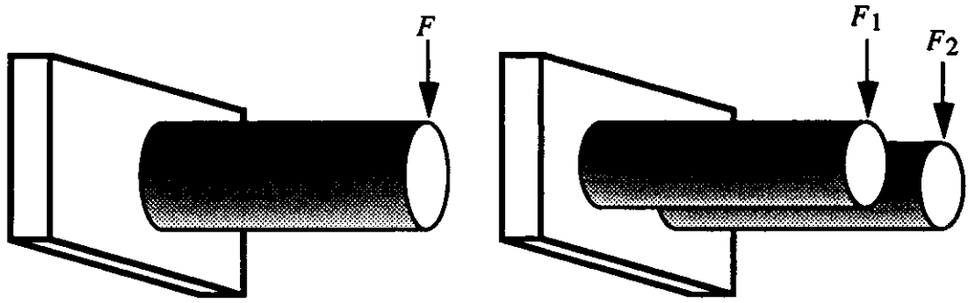

Figure 5. Application of IVE on a beam example

6. Search using Optimization

Optimization is an approach to problem solving in which an objective, $f(\mathbf{x})$, is minimized over a set of variables, $\mathbf{x}$, bounded by a set of equality and inequality constraints, $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ and $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. Optimization can be used in conjunction with our knowledge representation and expansion library in two ways: (a) to find the optimal solution within each design topology and (b) to determine which expansion technique to apply for expanding the current design topology. These roles of optimization are discussed in the two next sections.

6.1 Search for an Optimal Design within a Design Topology

The traditional role of optimization in engineering design is that of sizing individual components [Hrymak, 1985]. A more modern use of optimization is the simultaneous selection of a design topology and the sizing of the components within this topology. The problem definition makes use of a superstructure of alternative design topologies, i.e. a representation where multiple design topologies are imbedded [Duran and Grossmann, 1986]. This form of optimization employs the use of binary variables, whose values indicate the existence or non-existence of a particular component, and continuous variables, whose values indicate sizes of individual components and operating conditions.

Both of the described uses of optimization are applicable to predefined design topologies. The challenge in this class of problems lies in locating the global optimum within a reasonable amount of time. Research in this field continues in the direction of improved search algorithms, more efficient problem formulations, and global optimization [Nemhauser, *et al.*, 1989].

Both symbolic and numerical optimization techniques are available. We have employed the symbolic technique of monotonicity analysis [Papalambros and Wilde, 1988] to reason about functional relations between variables. In monotonicity analysis only the direction of the rate of change of one variable with respect to another is necessary. Monotonic functions have a constant sign in all their first partial derivatives. The level of abstraction in monotonicity

analysis is similar to those found in the traditional qualitative reasoning literature [Forbus, 1983; de Kleer and Brown, 1983; Williams, 1991]. Numerical optimization can be also applied to reason about functional relations in designs that are too complex for optimization using monotonicity analysis.

6.2 Optimization and Design Space Expansion

The design space expansion techniques of DVE and IVE are applicable to all dimensional and all input variables respectively. Among all possible expansions, only expansions of regions with critical variables will produce improved designs. *Critical variables* are those which influence the objective, and whose expansion will also influence the objective.

Cagan [1990] has proven a necessary condition for improving the objective after design space expansion for a certain class of problems. The *Rule of Critical Variables* is strictly applicable to initial and resulting primitive-prototypes that involve only monotonic constraints and objectives. It states that critical variables always appear in *active constraints* in monotonic problems. Active constraints are those which influence the location of the optimum in a design. The *Rule of Critical Variables* is applicable to both DVE and IVE, but these techniques do not require monotonic functions. The rule can still be applied in a heuristic manner in problems involving non-monotonic functions, and still identifies critical variables. Symbolic and numerical optimization is used to identify active constraints, and therefore candidate critical variables.

7. Limiting Designs by Induction

Repeated application of design space expansion and subsequent optimization of the resulting primitive-prototype frequently reveals patterns of interesting constraint activity. After each expansion, monotonicity analysis or numerical optimization derives sets of active constraints and an optimization analysis of the design is performed. This sequence of actions is called a *design generation*. If the activity of the analogous constraints remains the same across several design generations, then we can induce that this pattern will hold for an infinite number of generations, and take the limit of this expansion activity. More formally, Cagan and Agogino [1991a] define:

> *Inductively active (inactive) constraint*: If a constraint is active (inactive) for n consecutive generations of expansion then it is induced to be active (inactive) for infinite generations.

The number of consecutive design generations, n, required before attempting induction is specified by the user.

Induction is not a required step, rather it is an additional analysis step capable of produc-

ing certain designs which would otherwise require infinite *design generations*. Induction is fundamentally an aggressive design policy, which risks the possibility that some constraint be violated at the limit. It is important to check the result of induction against the design constraints to ensure that they have not been violated. In addition, when taking limits a designer should always determine whether or not the primitive-prototype remains a valid model of the design.

8. 1stPRINCE Design Methodology

The first principle knowledge representation, the library of expansion techniques, the optimization-based search, and the induction step have been incorporated in the 1stPRINCE design methodology. 1stPRINCE is an acronym for FIRST PRINnciple Computational Evaluator. The goal of this methodology, introduced by Cagan and Agogino [1987, 1991a, 1991b] and extended by Aelion, *et al.* [1991 and 1992], has been to innovate engineering designs. The means for innovating designs is the introduction of *new features* in a known design, by applying design space expansion techniques.

In 1stPRINCE, expansion techniques operate on a space of design solutions, defined by first principle knowledge. This knowledge, described in section 4, is captured in a primitive-prototype, which consists of an algebraic objective, algebraic equality and inequality constraints, a design topology, and constraint and variable types. An optimum is determined within the space of solutions, and the space is expanded (augmented). This new design space, while including the previous designs, provides an opportunity for including better designs.

The design space in 1stPRINCE changes dynamically, and goes beyond constraint satisfaction to produce designs which approach the global minimum. Cagan and Agogino [1991b] describe this design process as *optimally directed*, defined as an approach to design which attempts to determine optimal regions of the design space by directing the search toward improving the objectives and eliminating suboptimal regions.

1stPRINCE is described conceptually in Figure 6. The initial design is optimized. If the resulting design meets the designer's requirements a satisfactory solution is found and the design is completed. Alternatively, the design space is expanded via the library of design space expansion techniques. The resulting design is again subjected to optimization and the design loop is repeated. Four points are of particular interest:

(i) If specific design trends appear after a certain number of iterations, then the corresponding limit is induced to obtain the maximum benefit of that expansion.

(ii) If expansion does not improve performance, the design generation is stopped. Performance can be enhanced either by improving the objective or by satisfying previously violated constraints.

(iii) This algorithm makes use of optimization in two ways: (a) the traditional way of optimizing a given design and (b) the use of optimization information to suggest which expansion may improve design performance. The results are optimally directed design configurations.

(iv) The designer is involved in each design generation in two ways: (a) in deciding if a design is satisfactory and (b) in choosing among candidate expansion techniques determined algorithmically.

1stPRINCE is a domain independent approach to innovation in design, based on a mathematical representation. We have presently used this approach to create new design alternatives in the domains of chemical engineering, mechanical engineering and economics.

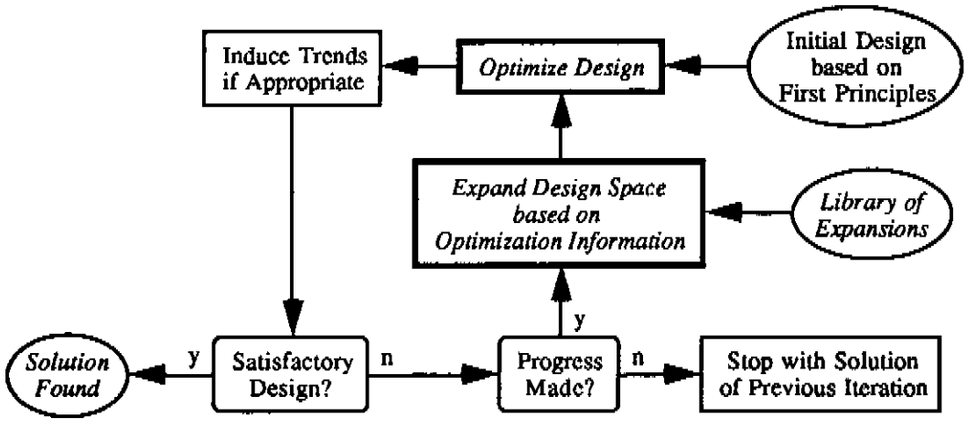

Figure 6. The 1stPRINCE design methodology

9. Algorithm and Implementation

The 1stPRINCE algorithm, shown in Figure 7a, examines the presence of candidate critical dimensional and/or input variables and reports candidate expansion techniques. The designer specifies one of these techniques for the next design generation. The algorithms for DVE and IVE are shown in Figures 7b and 7c respectively. In some cases we can determine *a priori* which expansion technique will not produce better designs; however, we cannot predetermine which technique, or combination of techniques, will produce the most improvement. Currently the designer chooses among candidate techniques (step 5 in Figure 7a) and the resulting objective is checked for improvement over that of the previous generation.

The DVE and IVE modules of 1stPRINCE have been implemented in Mathematica on a Macintosh IIcx. The induction module has been implemented in Franz Lisp and Flavors on Vax-series computers by Cagan and Agogino [1991a]. Monotonicity analysis, symbolic algebra, and symbolic application of the optimality conditions have been implemented in the SYMON/SYMFUNE programs by Choy and Agogino [1986] and Agogino and Almgren [1987]. The user must currently input the problem formulation into each module. The methodology remains to be integrated into a single environment.

BEGIN (*INPUT *primitive-prototype**)
1. Specify model template (objective and constraints).
2. Specify regions of initial design (topology).
3. Specify variable and constraint types.

END
BEGIN (*1stPRINCE*)
4. If first design generation, go to step 8.
5. Select critical variable and expansion type.
6. Choose a number of expansion regions (default = 2).
7. Update design topology by calling expansion technique.
8. Apply assignment constraints.
9. Create new boundary constraints.
10. For each region create parallel constraints.
11. Update objective function and serial constraints.
12. Optimize design.
13. If design generations > induction limit, and analogous constraints remain active (inactive), then induce expansion limit.
14. If design satisfactory, return result.
15. If design improved, go to step 5.
16. Return current design.

END

Figure 7a. The 1stPRINCE algorithm

BEGIN (*DVE*)
1. R = region involving critical dimensional variable.
   RU = connected regions topologically upstream of R.
   RD = connected regions topologically downstream of R.
2. Replace R with specified number of connected expansion regions.
3. Connect uppermost expansion region with all regions previously connecting RU and R.
4. Connect lowermost expansion region with all regions previously connecting R and RD.
5. Return new topology.

END

Figure 7b. The DVE algorithm

```
BEGIN (*IVE*)
  1. R = region involving critical input variable.
     RU = connected regions topologically upstream of R.
     RD = connected regions topologically downstream of R.
  2. Replace R with specified number of unconnected
     expansion regions.
  3. Connect all expansion regions with all regions
     previously connecting RU and R.
  4. Connect all expansion regions with all regions
     previously connecting R and RD.
  5. Return new topology.
END
```

Figure 7c. The IVE algorithm

10. Examples

The following examples aim to demonstrate the power of innovation and the domain independence of the 1stPRINCE design methodology.

10.1 Innovation of a Vehicle Fueling Design

Consider a problem where the net weekly income of a shipping process is to be maximized. The income depends on the annual volume of freight carried and fuel costs. The vehicle has volume capacity, Q. The volume of the merchandise is x. The total number of trips per week is v/d, where v is the average speed and d is the shipping distance. Fuel consumption varies with the square of the speed and the distance, βdv^2, where β is a constant. The weekly fuel consumption, βv^3, is equal to the fuel consumption in each trip, βdv^2, times the number of trips, v/d. P is the revenue per unit volume of delivered merchandise and F is the cost per unit volume of fuel. The problem of minimizing net weekly cost can be written as:

$$\begin{aligned} &\text{Minimize} && C - R \\ &\text{Subject to} && R = Pxv/d \\ & && C = \beta F v^3 \\ & && x + \beta dv^2 \le Q, \end{aligned}$$

where R and C are the total revenue per week and the total cost per week respectively. One possible input to 1stPRINCE includes variable and constraint typing:

Minimize	$C - R$	(f_1 - objective)
Subject to	$R = Pxv/d$	(a_1 - assignment constraint)
	$C = \beta F v^3$	(a_2 - assignment constraint)
	$d_1 = d$	(a_3 - assignment constraint)
	$x + \beta d_1 v^2 \leq Q$	(p_1 - parallel constraint)

The variable type specifications are given below:

R = total revenue / week (assignment),
C = total cost / week (assignment),
x = merchandise volume / trip (system, input),
d_1 = region distance per trip (system, dimensional),
d = total distance per trip (assignment),
v = speed (region).

Constraints (a_1) and (a_2) define the total revenue / trip and the total cost / trip respectively. Constraint (a_3) specifies that the trip distance in the design region is equal to the total trip distance, d, because there is only one design region in this design. Finally, constraint (p_1), the only parallel constraint in this design, specifies that the sum of the volumes occupied by the merchandise and the fuel cannot exceed the total volume of the vehicle.

1stPRINCE starts by optimizing the initial design, shown in Figure 8. The shaded area represents the fraction of the total capacity used for storing merchandise, and the remaining is the fuel volume. Monotonicity analysis indicates that all constraints are active and constraint (l_1) is satisfied as an equality. A detailed application of monotonicity analysis within 1stPRINCE appears in [Aelion, *et al.*, 1992]. By optimization and backsubstitution of the active constraints into the objective function we determine that the minimum net weekly cost is:

$$f_{1,\,\min} = \frac{-2\,P\,Q^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}}\left(\frac{P}{P+F}\right)^{1/2}.$$

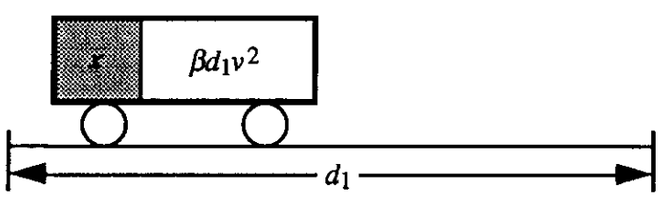

Figure 8. First Generation of the Fueling Design

The designer may either accept the above design, or search for a better design. The designer can do the latter by directing 1stPRINCE to apply DVE. The only dimensional variable is the distance, d_1. The region which contains d_1 is expanded to the default value of two regions. The corresponding design is expressed with the following primitive-prototype:

Minimize	$C - R$	(f_2 - objective)
Subject to	$R = Pxv/d$	(a_1 - assignment constraint)
	$C = \beta F v^3$	(a_2 - assignment constraint)
	$d_{11} + d_{12} = d$	(a_3 - assignment constraint)
	$x + \beta d_{11} v^2 \leq Q$	(p_{11} - parallel constraint)
	$x + \beta d_{12} v^2 \leq Q$	(p_{12} - parallel constraint)

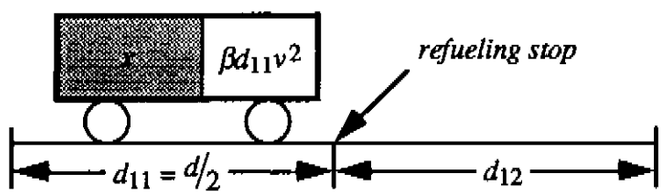

Figure 9. Second Generation of the Fueling Design

In this two-region design, shown in Figure 9, constraint (a_3) specifies that the sum of the distances in each region be equal to the total distance of the trip. Again, all constraints are active and the parallel constraints, (p_{11}) and (p_{12}), are satisfied as equalities. After backsubstitution and symbolic optimization, it turns out that d_{11} is equal to d_{12} and the minimum weekly cost of this design becomes:

$$f_{2,\,\min} = \frac{-2\,P\,Q^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}} \left(\frac{P}{\frac{P}{2} + F} \right)^{1/2}.$$

This objective, $f_{2,\,\min}$, is smaller than the previous objective, $f_{1,\,\min}$, indicating an improvement in performance. 1stPRINCE has been able to innovate because it has introduced a new feature into the original design. The new feature, *stopping to refuel*, provides a larger fraction of the vehicle capacity for storing merchandise, as shown by the increase of the shaded area in Figure 9. The stop was created by recognizing the existence of a dimensional variable and applying DVE to the design space.

The same procedure can be carried through for another design iteration to determine that

the four-region design produces further improvement. The objective becomes:

$$f_{4,\,\min} = \frac{-2\,P\,Q^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}}\left(\frac{P}{\frac{P}{4}+F}\right)^{1/2}.$$

1stPRINCE now notes that the analogous constraints remain active across design generations and induces the limit of infinitely many and differentially small design regions which sum up to the total distance of the trip, d. The objective becomes:

$$f_{n,\,\min} = \frac{-2\,P\,Q^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}}\left(\frac{P}{\frac{P}{n}+F}\right)^{1/2} \Rightarrow \lim_{n\to\infty}[f_{n,\,\min}] = \frac{-2\,P^{3/2}\,Q^{3/2}}{3^{3/2}\,d^{3/2}\,\beta^{1/2}\,F^{1/2}}.$$

This objective indicates the maximum improvement that DVE can produce with this system description. 1stPRINCE has been able to innovate the concept of a vehicle which refuels continuously along a trip, shown in Figure 10, and resembles the operation of an electric bus. In this design, all the volume capacity of the vehicle is used for storing merchandise. This innovation happens automatically as 1stPRINCE manipulates the initial design using its library of heuristic expansions. The designer also uses 1stPRINCE to try IVE, which proposes the design of two vehicles making this trip in parallel. Given the initial system description, this design does not improve the objective.

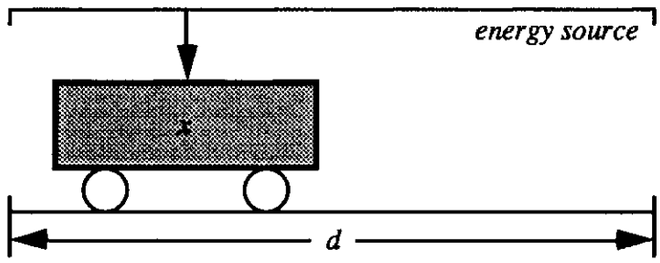

Figure 10. The Fueling Design after Induction of Constraint Activity

In this formulation several simplifying assumptions have been made: the average speed of the vehicle, v, remains unaffected with the creation of refueling stops and there is are no costs associated with stopping to refuel, building refueling stops, or building an energy source. The problem can be solved without these assumptions and perhaps innovate still other designs. The main interest during conceptual design is to observe trends of improvement by expanding the design alternatives.

Ultimately, the designer must recognize the meaning and implement the design ideas produced by I-PRINCE. The application of this design methodology helped in exploring alternative designs and provided the stimulus of a new design idea. I-PRINCE works interactively with the designer who critiques the emerging designs and provides guidance on which expansion types to be applied. The automation of this methodology supports rapid exploration of a certain class of innovative designs.

The following examples are applications of I-PRINCE to mechanical and chemical engineering designs, which aim to demonstrate the domain independent nature of the approach.

10.2 Innovation in Beam Design

I-PRINCE was originally developed for mechanical design, and applied to various structural and dynamics models. In this example, the weight of a solid beam of circular cross-section under flexural or torsion load is to be minimized. In addition, the bending and shear stresses should not exceed the corresponding yield stresses. The design can be formulated as follows:

Minimize	weight
Subject to	bending stress < bending yield stress
	shear stress < shear yield stress

A detailed analysis of this design can be found in [Cagan and Agogino, 1991a]. Depending on the critical variable and the expansion technique, a number of alternative designs have been innovated, as shown in Figure 11. Depending on their complexity, most of these designs are obtained in closed form.

Application of DVE along the length of the beam under flexural load and subsequent independent modeling of each region, produces the concept of a composite beam. If the material properties of the individual regions are required to remain the same, optimization makes the radii progressively smaller, and induction innovates the concept of a tapered beam. DVE over the radius of a beam under torsion load, and independent densities of the resulting regions produces the concept of a hollow tube, because optimization drives the density of the internal region to zero. IVE over either flexural or torsion loads proposes the design of multiple beams, each of which carries only a fraction of the original load. This may be useful if it is impossible to achieve the design goal without violating the constraint. Finally, combinations of these techniques could produce solutions like the hollow / tapered beam shown in Figure 11. I-PRINCE has been able to innovate these designs by creating new features. Subsequent optimization gives these features the most desirable shape.

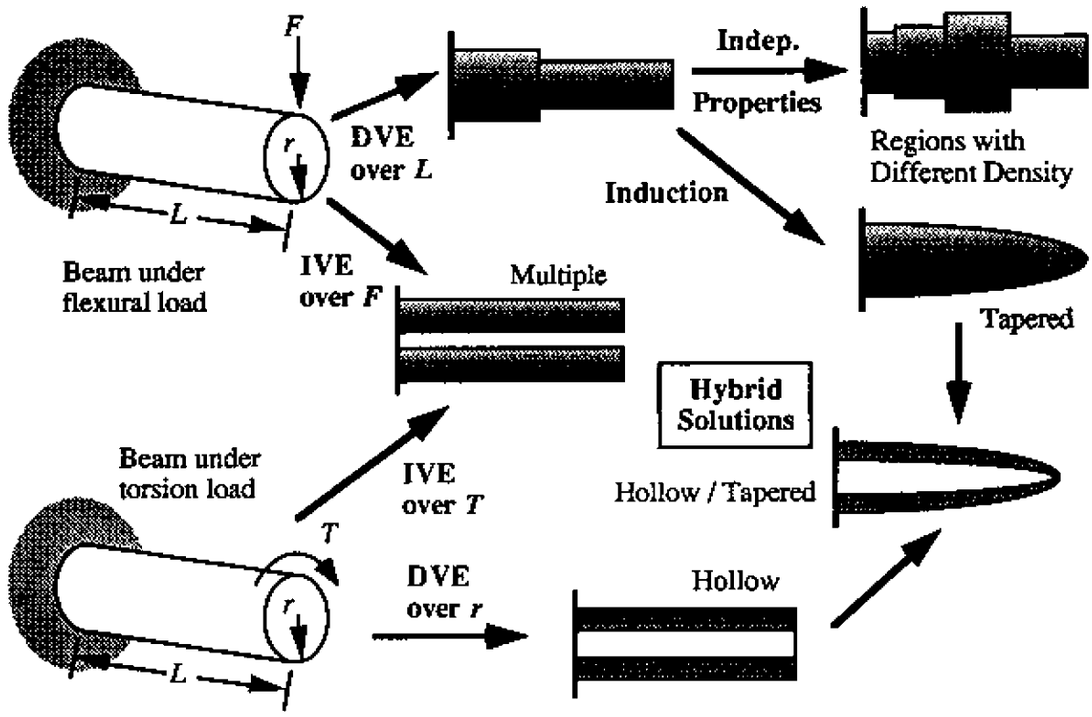

Figure 11. Application of I'PRINCE to beam design

10.3 Innovation of Chemical Reactor Systems

An application of I'PRINCE to the design of chemical reactors is shown in Figure 12. The goal of this design is to maximize a first-order chemical reaction, subject to a maximum volume constraint. The initial design is that of a *well-mixed reactor.* The well-mixed reactor is an ideal conception in which all the contents are perfectly mixed. The design becomes:

Minimize	exiting reactant concentration
Subject to	well-mixed reactor mass balance
	reactor volume < upper bound

A detailed analysis of this and similar problems can be found in [Aelion, *et al.,* 1991 and 1992]. Applications of DVE on reactor volume and IVE on input flow produce serial and parallel reactor systems. Of particular interest is the limit of infinitely many and differentially small well-mixed reactors in series, which produces the identical behavior of a *plug-flow reactor.* This reactor type is another theoretical conception, where there is no axial mixing. There is only radial mixing in every differential slug traversing the reactor. I'PRINCE is able to innovate this design based purely on variable and constraint typing and the application of DVE.

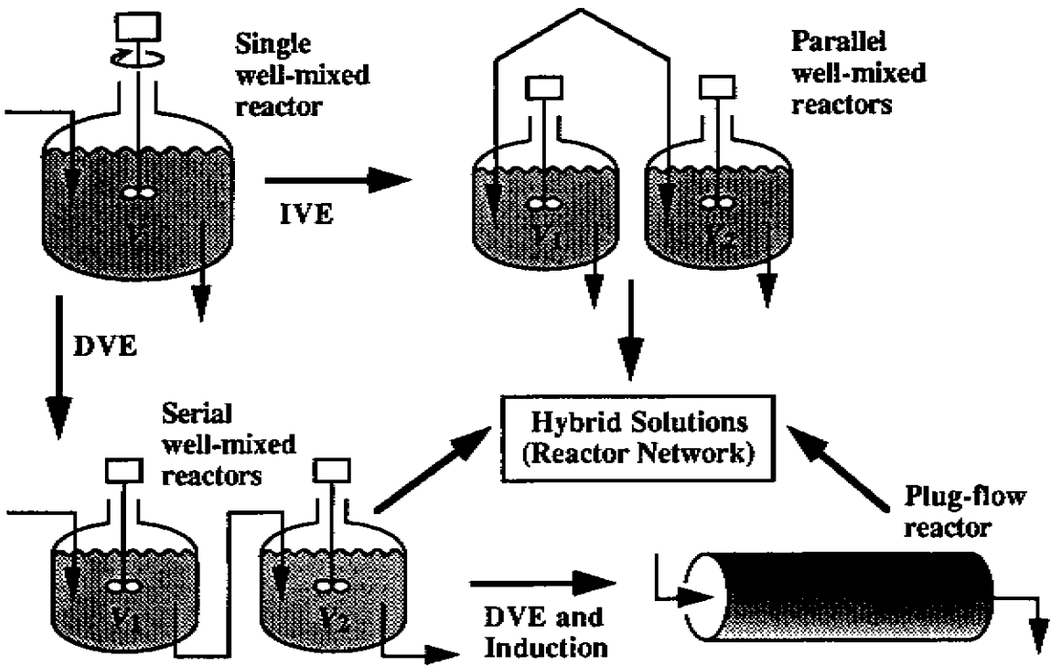

Figure 12. Application of 1stPRINCE to chemical reactor design

10.4 Reinventing the Wheel

1stPRINCE has been applied to the following problem:

Minimize	resistance to spinning
Subject to	object area ≥ lower bound

Starting from the initial design of a square, shown in Figure 13, application of DVE over two dimensions, while maintaining symmetry, has proposed the design of a wheel. This example is described in detail in [Cagan and Agogino, 1987].

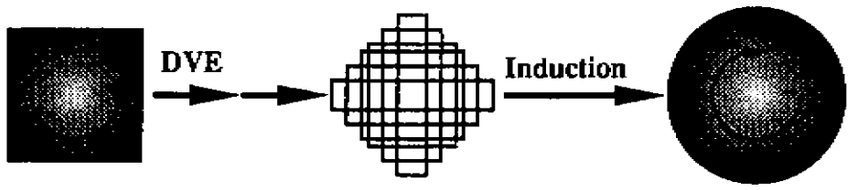

Figure 13. Application of 1stPRINCE to the minimum resistance design

11. Discussion

This paper describes (a) a knowledge representation which is based on first principles, (b) a library of mathematical heuristics for expanding the space of design alternatives, and (c) an optimization-based search technique. These have been combined in the 1stPRINCE methodology for innovation in engineering design.

The first principle knowledge representation, reviewed in section 4, is based on the engineering paradigm of modeling systems with mathematical relations. The representation also includes variable and constraint typing. Variable typing is domain dependent knowledge which describes fundamental roles of variables within a design. Constraint typing describes the region(s) in which each constraint is applicable in a design. Typing provides additional information on a system, which may be necessary for reasoning in particular applications. We have used this information to decide on how to extend the space of design alternatives. This representation may be useful in other applications that are amenable to mathematical representations and that require additional information on variable characteristics, constraint applicability, and topology to achieve their reasoning goals.

The expansion techniques, described in section 5, are based on the idea that introducing new independently modeled regions in designs can produce new features in these designs. Currently the library of expansion techniques has two members: DVE and IVE. These techniques are the grammar for modifying design topology, so the more members in this library, the stronger the power to create new designs. DVE and IVE have been defined as mathematical operators, a formulation which accommodates our knowledge representation. The concept of design space expansion, however, is useful independently of knowledge representation. In their current form, design space expansion techniques target variables with specific roles in a design. As we identify variables with other interesting roles, we can define corresponding variable types and develop corresponding expansion techniques.

The 1stPRINCE design methodology is a domain independent framework for innovation in engineering design. It starts with an initial design and innovates designs by adding new features. The methodology makes use of optimization in a unique way because it employs mathematics to perform synthesis, even though its traditional role is in analysis. 1stPRINCE also uses induction to examine the limiting designs for each expansion.

With the exception of designs produced by induction, the results of 1stPRINCE are *sound*, because the newly created regions are modeled by physically valid equality and inequality constraints (assuming a correct initial primitive-prototype). Induction, on the other hand, is an aggressive design policy which poses two risks: (a) some constraints may be violated at the limit, and (b) the primitive-prototype may or may not be a valid model of the substantially different topology. When induction has been used, the user must act as a critic of

the resulting designs. With this additional check, 1stPRINCE becomes a *sound* algorithm.

The generation of designs by 1stPRINCE is combinatoric. The system offers a choice among expansion types, target regions for expansion, and number of newly created regions. The combinatorial explosion can be managed in two ways: (a) automatically prune out certain expansions based on domain knowledge, and (b) interactively include the user in the design loop to help make decisions when the preferred course of action is ambiguous. In limited cases, currently we can assess *a priori* which expansion type or region will produce better designs.

1stPRINCE may be viewed as a part of a larger system for engineering design, where a front-end methodology would develop an initial design, which would be further processed by 1stPRINCE to produce other design alternatives. All these alternatives could then be incorporated into a superstructure to be evaluated in detail by mixed-integer non-linear programming (MINLP) optimization methods. [Williams, 1991] presents an interesting methodology for discovering topological configurations which could be inputs to 1stPRINCE.

12. Conclusions

A first principle knowledge representation for innovation in engineering design has been presented. It consists of an algebraic objective, algebraic equality and inequality constraints, and variable and constraint types. The constraints come from the fundamental physics and chemistry, as well as manufacturing, safety, and performance considerations. This representation is useful in domains which can be described mathematically, but additional information needs to be modeled for reasoning purposes.

A library of design space expansion techniques has been also presented. These techniques are used to innovate designs by introducing new features into known designs. Presently two techniques are available: *Dimensional Variable Expansion* and *Input Variable Expansion*. The concept of design space expansion can be useful independently of our design methodology.

The knowledge representation and the library of design space expansion techniques have been combined into the 1stPRINCE design methodology. This methodology uses optimization information to decide which expansion technique may produce improved designs and induction to examine limiting behavior. Starting from an initial design, 1stPRINCE has innovated interesting engineering concepts, such as the *electric bus*, the *tapered* and *hollow beam*, the *wheel*, and the *plug flow reactor*.

13. Acknowledgments

The authors would like to thank Leo Joskowicz, Tom Mitchell, David Steier, and Brian Williams for their comments on this manuscript.

14. References

Aelion, V., J. Cagan and G. J. Powers [1992]. Input Variable Expansion - An Algorithmic Design Generation Technique. In press: *Research in Engineering Design.*

Aelion, V., J. Cagan and G. J. Powers [1991]. Inducing Optimally Directed Innovative Designs from Chemical Engineering First Principles. *Computers & Chemical Engineering*, **15**, 9, 619-627.

Agogino, A. M., and A. S. Almgren [1987]. Techniques for Integrating Qualitative Reasoning and Symbolic Computation in Engineering Optimization, *Engineering Optimization*, **12**(2), 117-135.

Brown, D. and B. Chandrasekaran, [1985]. Expert Systems for a Class of Mechanical Design Activity. In J. Gero (ed.), *Knowledge Engineering in Computer-aided Design*, North Holland, Amsterdam.

Cagan, J. and A. M. Agogino [1987]. Innovative Design of Mechanical Structures from First Principles. *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, **1**(3), 169-189.

Cagan, J. and A. M. Agogino [1991a]. Inducing Constraint Activity in Innovative Design. *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, **5**(1), 47-61.

Cagan, J. and A. M. Agogino [1991b]. Dimensional Variable Expansion - A Formal Approach to Innovative Design. *Research in Engineering Design*, **3**, 75-85.

Cagan, J. [1990], *Innovative Design of Mechanical Structures from First Principles*, Ph.D. Dissertation, University of California at Berkeley, April 1990.

Choy, J. K. and A. M. Agogino [1986]. SYMON: Automated SYMbolic MONotonicity Analysis System for Qualitative Design Optimization. *Proceedings of ASME 1986 International Computers in Engineering Conference*, Chicago, 305-310.

de Kleer, J. and J. S. Brown [1983]. Qualitative Physics based on Confluences. *Artificial Intelligence*, **24**, 7-84.

Dcyer, M. G., M. Flowers, and J. Hodges [1986]. EDISON: An Engineering Design Invention System Operating Naively. In *Applications of Artificial Intelligence in Engineering Problems*, Springer-Verlag, 327-341.

Duran, M. A. and I. E. Grossmann [1986]. A Mixed Integer Nonlinear Programming Algorithm for Process Systems Synthesis. *AIChE J*, **32**, 592-606.

Faltings, B. [1991]. Qualitative Models in Conceptual Design: A Case Study. *Artificial Intelligence in Design '91* (Gero, J. S., *ed.*), Butterworth Heinemann, 645-663.

Forbus, K. [1983]. Qualitative Process Theory. *Artificial Intelligence*, **24**, 85-168.

Hrymak, A. N., G. J. McRae, and A. W. Westerberg [1985]. Combined Analysis and Optimization of Extended Heat Transfer Surfaces. *J. Heat Transfer*, **107**, 527-532.

Joskowicz, L. and S. Addanki [1988]. From Kinematics to Shape: An Approach to Innovative Design. *Proceedings of AAAI-88*, St. Paul, August 21-26, **1**, 347-352.

Lenat, D. B. [1983]. EURISKO: A Program that Learns New Heuristics and Domain Concepts. The Nature of Heuristics III: Program Design and Results. *Artificial Intelligence*, **21**, 61-98.

Lenat, D. B. [1982], The Nature of Heuristics. *Artificial Intelligence*, **19**, 189-249.

Lenat, D. B. [1983a], The Nature of Heuristics II: Theory Formulation by Heuristic Search. *Artificial Intelligence*, **21**, 31-59.

Maher, M. L., F. Zhao, and J. S. Gero [1989]. An Approach to Knowledge-Based Creative Design. *Proceedings of NSF Engineering Design Research Conference*, Amherst, MA, June 11-14, 333-346.

Mavrovouniotis, M. L. and G. Stephanopoulos [1988]. Formal Order-of-Magnitude Reasoning in Process Engineering. *Computers & Chemical Engineering*, **12**, 867-880.

McDermott, D. [1977]. Flexibility and Efficiency in a Computer Program for Designing Circuits. *MIT AI Lab TR-402*.

Mitchell, T. M., *et al.* [1983]. An Intelligent Aid for Circuit Redesign. *AAAI*.

Mitchell, T. M., S. Mahadevan, and L. I. Steinberg [1985]. LEAP: A Learning Apprentice for VLSI Design. *IJCAI*, 573-580.

Mitchell, W. J. [1989]. A Computational View of Design Creativity. In *Preprints of Modeling Creativity and Knowledge-Based Creative Design*, International Round-Table Conference, Heron Island Queensland, December 11-14, 263-285.

Mittal, S., *et al.* [1986]. PRIDE: An Expert System for the Design of Paper Handling Systems. *Computer*.

Murthy, S. S. and S. Addanki (1987). PROMPT: An Innovative Design Tool. *Proceedings of AAAI-87*, Seattle, WA, July 13-17, **2**, 637-642.

Nemhauser, G. L., A. H. G. Rinnooy Kan, and M. J. Todd (eds.) [1989]. *Handbooks in Operations Research and Management Science. Volume 1: Optimization.* Elsevier Science Publishing Company, NY.

Papalambros, P. and D. J. Wilde [1988]. *Principles of Optimal Design*. Cambridge University Press, Cambridge.

Ressler, A. [1984]. A Circuit Grammar for Operational Amplifier Design. *MIT AI Lab TR-807*.

Roylance, G. [1980]. A Simple Model for Circuit Design. *MIT AI Lab TR-703*.

Sandler, S. I. [1977]. *Chemical and Engineering Thermodynamics*. John Wiley, New York.

Stiny, G. [1980]. Introduction to Shape and Shape Grammars. *Environment and Planning B*, **7**, 343-351.

Tong, C. and D. Sriram (eds.) [1991]. *Artificial Intelligence Approaches to Engineering Design*. Addison-Wesley.

Ulrich, K. and W. P. Seering [1988]. Function Sharing in Mechanical Design. *Proceedings of AAAI-88*, St. Paul, August 21-26, **1**, 342-346.

Williams, B. C. [1990]. Invention from First Principles: An Overview. P. Winston and S. Shellard (eds.). *Artificial Intelligence at MIT: Expanding Frontiers*. MIT Press, Cambridge, MA.

Williams, B. C. [1991]. A Theory of Interactions: Unifying Qualitative and Quantitative Symbolic Algebra. *Artificial Intelligence*, **51**, 39-94.