# Input Variable Expansion
# A Formal Innovative Design Generation Technique

Vital Aelion, Jonathan Cagan, Gary J. Powers

EDRC 24-62-91

# Input Variable Expansion
## A Formal Innovative Design Generation Technique

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

*Input Variable Expansion,* **IVE, is a new domain independent formal methodology for creating innovative designs. These designs are based on a known design which is cast as an optimization problem described by its first principle equations. IVE performs design space expansion by replicating the topology of the initial design, assigning independent properties to each region and distributing a selected input to the newly created regions. Optimization information is employed in the selection of the distributed input.**

**The resulting design is optimized, using symbolic optimization techniques when possible. In more complex and industrially relevant problems where symbolic methods are more difficult, numerical methods are used to optimize the resulting designs. Trends over generations of designs are observed and the limiting designs are induced. The innovated designs may exhibit either an improved objective or a feasible design space replacing an infeasible one.**

**IVE is a complementary expansion technique to** ***Dimensional Variable Expansion,*** **DVE, developed by Cagan and Agogino (1991a). Together, IVE and DVE initiate a library of design space expansion techniques which, in some cases, eliminate the need for prepostulated superstructures for finding the optimal solution. IVE is demonstrated in applications to the innovative designs of a catalyst bed, a set of columns under axial load and a chemical reactor network.**

## 1. Introduction

Traditionally, design concepts are optimized by improving the performance of an initial design over its continuous variables (Reklaitis *et. al.*, 1983; Papalambros and Wilde, 1988). Alternatively, innovative designs often result from expanding the design space by generating alternative features for achieving the design concept. For example, in the design of a chemical reactor system to convert a feed stream into a desired product, the optimal feed composition, flowrate and reactor volume could be determined by optimizing over these variables. Innovation in this design space might involve multiple reactors in serial and/or parallel configurations, different types of reactors, etc. The generation and selection of alternative design structures is a limiting step in design innovation and optimization (Westerberg, 1989).

Design alternatives are often synthesized by modifying and improving known designs of related functionality. This design transformation is frequently accomplished by applying domain specific heuristic knowledge which embodies the designers[1] understanding of the sensitivities and interactions of the design variables and parameters (Lenat, 1983; Murthy and Addanki, 1987; Ulrich and Seering, 1988; Joskowitz and Addanki, 1988). Other design transformations can be achieved through the application of genetic algorithms (Androulakis and Venkatasubramanian, 1991). This paper advocates that known designs can be improved by directly applying optimally directed design space expansion techniques on first principle information (i.e. the algebraic equalities and inequalities developed from the fundamental physics, chemistry, manufacturing and safety constraints of a system).

*Input Variable Expansion,* IVE, is a formal technique capable of generating new designs by expanding the *input variables* of an initial design in an optimally directed manner. *Optimally directed design* is 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. This approach reduces the size of the search space and generates insight as to the desirable directions for improving the design variables (Cagan and Agogino, 1991a). The initial design is described by its first principle equations. Application of IVE results in an expanded design space, a requirement for producing *innovative designs.* Cagan and Agogino define innovative designs as those which involve new design variables or features based on variables or features of an existing design prototype. The basis of this technique is an initial design cast as an optimization problem, rendering IVE domain independent. The extent of required domain information is limited to specifying the nature of the variables involved. IVE innovates by either producing designs which outperformthe objective of the initial design, or by creating a feasible design space starting from an initially infeasible design.

IVE expands the design space by identifying critical input design variables (defined in subsection 2.3), dividing the design space across those variables, and making the properties within each newly-formed region independent from those of other regions. This manipulation frequently increases the *degrees-of-freedom* of the design space, which indicates potential innovation in the new designs. Optimization information is used in selecting the critical input variables, and at every iteration of the design expansion the resulting design is optimized. If trends of improvement appear when iteratively expanding an input variable with IVE, induction is used to generalize these trends, thus achieving the limit of improvement with respect to that particular variable. New designs are generated from the fundamental equations that describe a previously known design, so the need for prepostulated superstructures (i.e. domain descriptions that imbed several design alternatives, as described in Duran and Grossmann, 1986) for representing the design space is reduced.

Cagan and Agogino (1991a) have presented *Dimensional Variable Expansion,* DVE, another formal technique for design space expansion. DVE focuses its attention on variables that describe the physical dimensions of a design. DVE differs from IVE in that critical dimensional variables are selected for expansion instead of input variables. In so doing, DVE creates discontinuities in the initial design topology and permits property independence across these discontinuities. In contrast, IVE creates replicates of the initial design topology in which the selected input is distributed. The result is a unique, domain independent, formal approach to design space expansion. Because IVE and DVE are complementary processes with different design effects, their application may each be superior in different problems. Together DVE and IVE initiate a library of formal techniques for design space expansion. Both techniques are used within the general framework of the 1stPRINCE design methodology, initially presented in Cagan and Agogino (1987).

In this paper, the formal theory of IVE will be presented. The theory will be illustrated in a sequence of examples, followed by a discussion indicating reasons for its success and anticipated limitations.

## 2. Input Variable Expansion (IVE)

IVE is a formal technique for expanding the design space of a known design, called a *primitive-prototype,* in an effort to produce *better* designs. Conceptually IVE proposes the *parallel processing* of an input into design regions whose topologies are replicates of an initial design, as shown in Figure 1. The next subsection describes the concept *of a primitive-prototype.*

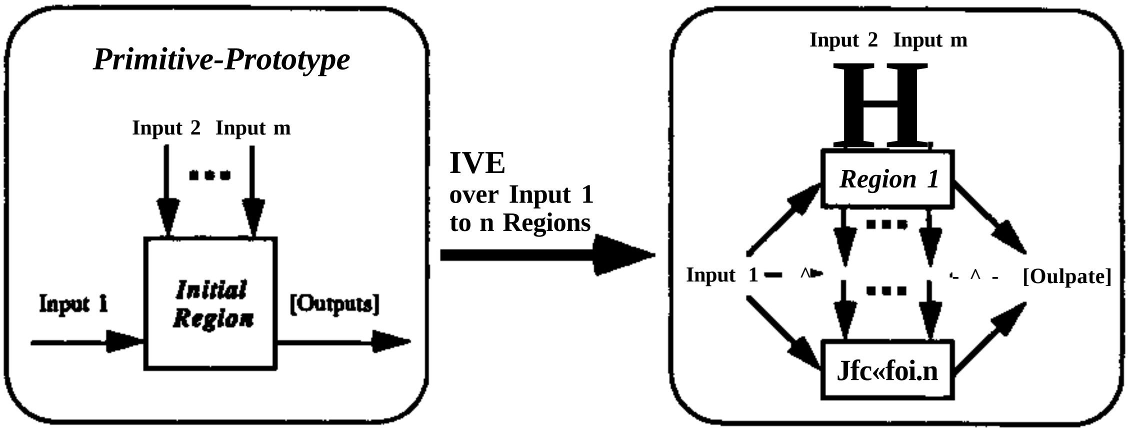



**Figure** 1. Conceptual illustration of IVE

## 2.1. Definition of a Primitive-Prototype

A *primitive-prototype* is defined by an objective function, /(x), and a set of variables, x, bounded by a set of equality and inequality constraints, h(x) = 0 and g(x) £ 0:

| | |
|---|---|
| *Minimize* | /(x) |
| *Subject to* | h(x) = 0 |
| | $g(x) \leq 0$. |

The primitive-prototype can be subdivided into a number of design *regions,* each of which is independently modelled with variables and constraints (Cagan and Agogino, 1991a).

We define five types of variables which characterize the way the variables are used to model the design. The first two, namely the *extensive* and *intensive* variables, designate whether these variables depend on the physical size of the system. The remaining three variables are subsets of these types which describe how IVE and DVE expand the design space.

*Extensive variables, x?*, express quantities that depend on a characteristic size of a system. Examples of such variables include weight, reactor volume, capacitor charge and mass flowrate. Extensive variables are replicated when new regions appear in the design. These variables have been defined previously by Cagan and Agogino as *system variables.*

*Intensive variables,* $x^{\dagger}$, express quantities that describe a property of a design region. Such quantities are independent of all the region characteristic sizes. Intensive variables include temperature, pressure, density, Young's modulus and thermal conductivity. Intensive variables are replicated when new regions appear in the design. These variables were defined by Cagan and Agogino as *region variables.* The definitions of extensive and intensive variables have been adopted from the field of thermodynamics; for example see Sandier (1977).

*Assignment variables,* $x^*$, express quantities that have contributions from multiple design regions. Examples include total mass of an object, total sales and total cost. Assignment variables are not expanded with the creation of new design regions. Rather, the new regions contribute a part to the value of an assignment variable specified by a global constraint (defined below).

*Dimensional variables,* ***, constitute a special subclass of extensive variables that denotes the physical dimensions of a design. Such variables have been the focus of DVE, as discussed in subsection 2.4.

*Input variables** $\dot{x}^m v$^\ form another subclass of extensive variables which describes *quantities to be processed* by a design. Examples of input variables include molar flows in chemical reactor designs and loads in structural designs. Input variables are the main focus of IVE where the design expansion is based on a designated input which is distributed to each individual design region. *Independent* extensive variables are selected as input variables. For example, in a reactor design, from the variable set {volumetric flowrate, mass flowrate, molar flowrate} only one can become an input variable because all these flowrates depend on the rate of material flow.

These variables appear in the design objective or constraints. Constraints are classified as *global, c&,* and *local,* $c^l$ (previously defined as *serial* and *parallel,* respectively, by Cagan and Agogino). Global constraints apply across all design regions. They are each modified to include contributions from each newly created region, according to a problem specific *update formula.* Update formulae usually designate a summation of a quantity across the various regions and may depend on the expansion technique. All the remaining constraints are local and apply to individual design regions. These local constraints are replicated with the creation of new design regions.

Based on these definitions, the primitive-prototype becomes:

*primitive-prototype = f bounded* by *[c&]* u {$c^1$} over *[x?]* u {$JC^1$} u {$x^a$} .

Dimensional and input variables are included in the set of extensive variables. A related definition from Cagan and Agogino is that of *& prototype,* a class of designs that result from an optimization analysis of a primitive-prototype and can be instantiated to at least one feasible design solution.

The next subsection addresses *constraint activity,* an instrumental concept for choosing candidate inputs for IVE.

## 2.2. Constraint Activity

**Design space expansion techniques interact with optimization in two ways: (a) optimization information is used to select target variables for expansion, and (b) the resulting primitive-prototype is optimized. A further description of this interaction appears in section 3.**

**Constraint activity is a form of optimization information which supports the selection of *critical variables,* discussed in the next subsection. An *active* constraint is defined by Papalambros and Wilde (1988) as one which, if removed, would alter the location of the optimum. If the problem is monotonic, active constraints are satisfied as equalities, allowing for a *degree-of-freedom* analysis of the IVE strategy. The IVE algorithm is described next.**

## 2.3. IVE Algorithm

**IVE, illustrated in Figure 1, expands the design space along a *critical* input variable, i.e. one which influences the objective function and which, when expanded, will create new variables which will also influence the expanded objective function (Cagan and Agogino, 1991a). All variables which appear in the objective and the possible sets of active constraints are potentially critical. Among those, the variables which represent inputs are candidates for expansion. IVE is realized by the algorithm shown in Figure 2.**

**BEGIN** (*INPUT*)
1. Formulate design problem as an optimization problem.
2. Specify the *input, extensive, intensive,* and *assignment* variables.
3. Present constraints in *global* and *local* form.
4. Specify an *update formula* for the objective and each global constraint.

**END**
**BEGIN** (*IVE*)
5. Identify a critical input variable.
6. **FOR** each existing region:
   **BEGIN**
   6a. Choose a number of replicate regions (default = 2).
   6b. Replicate extensive variables, intensive variables and local constraints to each new region.
   **END**
7. Modify objective and global constraints according to their update formulas.

**END**

**Figure 2. The IVE algorithm**

**Most frequently in designs, multiple critical dimensional and input variables exist, each of which is a candidate for DVE and IVE respectively. IVE can be repeated for every input variable separately, it can be superimposed on top of designs which have resulted from other**

IVEs, or it can be applied to designs which have resulted from the application of DVE. The result of all these possibilities depends on the coupling of the critical variables.

## 2.4. DVE Versus IVE

The DVE algorithm is described in Cagan and Agogino (1991a). The main differences between DVE and IVE are: (a) DVE focuses on a critical dimensional variable, as opposed to IVE which focuses on a critical input variable, and (b) in DVE the expanded regions are serially related to each other via imposed boundary conditions, whereas in IVE the expanded regions are parallel and no similar boundary conditions are imposed.

## 2.5. Degree-of-Freedom (DOF) Expansion in IVE

An increase in the *degree-of-freedom, DOF,* of a problem is an indication of potential innovation. IVE expands the design into multiple parallel regions, each of which exhibits the same topology as the initial design, and the resulting design is optimized. IVE creates new regions which involve new constraints and variables and support the possibility of realizing a superior design. At optimality a subset of the constraints is active, meaning that their position constrains the location of the optimum.

Consider the *DOF* analysis of a design described by a monotonic objective and monotonic constraints. In this case the active constraints are satisfied as equalities and the *DOF* is defined to be:

$$DOF = V - A\ , \tag{1}$$

where A is the number of non-redundant active constraints and V is the number of variables found in these constraints. Applying this to a primitive-prototype we get:

$$DOF_{p\text{-}p} = v^e + v* + v^a - u8 - u^1, \tag{2}$$

where $DOF_{P\text{-}P}$ is the *DOF* of the primitive-prototype, $v^e$, $v^1$ and $v^a$ are the numbers of extensive, intensive and assignment variables, and u£ and $u^1$ are the numbers of active global and local constraints, respectively, in the primitive-prototype. Application of IVE expands the primitive-prototype to *n* regions. The extensive and intensive variables and the local constraints are replicated in each of these regions. The global constraints are not replicated but are modified via their update formulae. The assignment variables are not expanded at all. In general, IVE may form a new primitive-prototype with a higher or lower *DOF* than the previous primitive-prototype, because each could be bounded by a different set of constraints. However, if after expansion the same dominant constraint activity is maintained as before the expansion, then the following results apply. *Dominant constraint activity* means that a constraint in a new prototype has the same activity as the analogous constraint from which it was derived in the old primitive-prototype (Cagan, 1990). For single-input designs, the resulting primitive-prototype exhibits the following *DOF:*

$$DOF_n = n\,(v^1 + v^e) + v^a - u« - n\,u^l = DOF_o + (n - 1)\,(v^* + v^e - u^1), \quad (3)$$

where $DOF_O$ and $DOF_n$ are the *DOFs* of the old and the new primitive-prototypes.

Equation (3) indicates that *the new DOF increases as long as the sum of the intensive and extensive variables exceeds the number of active local constraints in the old primitive-prototype.* If some or all of the regions of the new primitive-prototype are further expanded to multiple regions and dominant constraint activity is maintained, the *DOF* increases further, as indicated by applying the results indicated above for each region.

The *DOF* increase in DVE, £>0FDVE> is smaller than in IVE, DOFIVE, because in DVE the expanded regions are related through specified boundary conditions, as indicated in equation (4), which is derived by Cagan (1990):

$$DOF_{DVE} = v^e + v^* + v^a - u« - u^1 - \{n - 1). \quad (4)$$

Therefore if we attempt DVE and IVE on the same primitive-prototype, it follows that:

$$DOF_{IVE} = DOFDVE + 0*" \,!)\bullet \quad (5)$$

This analysis is valid for designs with monotonic objective and constraints, and a single input variable. In general, IVE is applicable to designs with non-monotonic objective and constraints, as well as multiple input variables. However, a different *DOF* analysis applies in these cases. The next section describes the interaction of IVE and optimization techniques.

## 3. IVE and Optimization

The design expansion performed by IVE is followed by the optimization of the resulting primitive-prototype. The overall design strategy is shown in Figure 3.

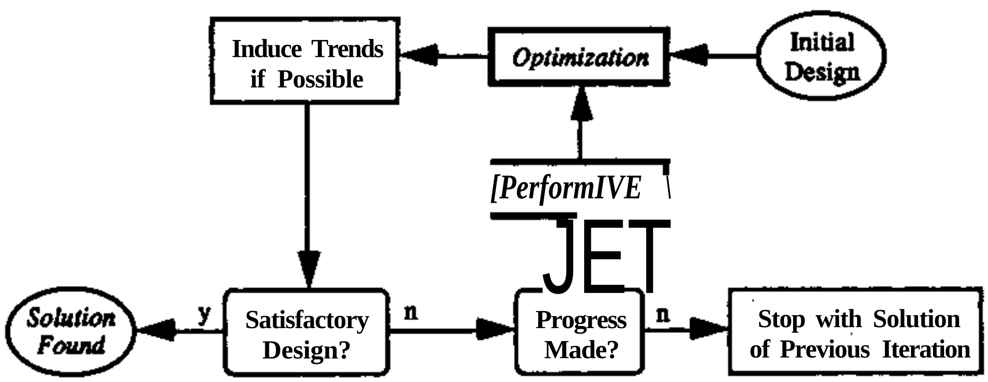



Figure 3. Overall Design Strategy

The initial primitive-prototype is optimized. If the resulting prototype meets the designer's requirements a satisfactory solution is found and the design is completed. Alternatively, IVE expands the design space and the resulting primitive-prototype is again subjected to optimization and the design loop is repeated. Two points ate of particular interest:

(a) 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. A further discussion on induction appears in the next section.

(b) If the application of IVE does not improve performance, the design iteration is stopped. Design performance can be increased either by improving the design objective or by satisfying previously violated constraints.

The design is carried out at the conceptual stage where detailed parametric information may not be available. Symbolic optimization is preferable when possible for addressing this lack of detailed information and also for gaining a better qualitative understanding of the design. However, numerical optimization may be required.

Monotonicity analysis, presented by Papalambros and Wilde (1988), is a methodology which can aid in obtaining symbolic solutions by analyzing the *boundedness* of the problem, i.e. whether the problem is well-constrained. A detailed application of monotonicity analysis in a design problem appears in Aelion *et. al.* (1991).

## 4. Induction of Constraint Activity in IVE

The repeated application of IVE and the subsequent optimization of the resulting primitive-prototype sometimes reveals patterns of constraint activity worth investigating. After each IVE application, monotonicity analysis derives sets of active constraints and a new optimization analysis is performed. This sequence of actions is termed 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 IVE applications, and take the limit of this expansion activity. More formally, Cagan and Agogino (1991b) 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 such generations.

The number of consecutive design generations, n, required before attempting induction in IVE is specified by the user. Its default value is 3.

Induction is not a required step, rather it is an additional analysis step capable of producing certain designs which would otherwise require infinite time. The act of induction is fundamentally an aggressive design policy, which risks the possibility that some constraint be violated at the limit. Therefore it is imperative that the result of induction be checked 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 prototype remains a valid representation of the underlying physics, chemistry and economics of the design.

The following examples have been selected to reveal the key features of IVE. For clarity, they represent only partial statements of industrially relevant design problems.

## 5. Example: Catalyst Weight Minimization

Consider the design of a catalyst whose total weight must be kept to a minimum, subject to a minimum total surface area constraint, which is required for supporting the reaction of interest. Step 1 of the IVE algorithm, shown in Figure 2, requires that the design be stated as an optimization problem. Assuming the primitive-prototype of a single spherical catalyst pellet, this design becomes:

*Minimize* *W*tot = P 'not

*w.r.t.* (Atot> ^i> ^l> $^r l$)

| | | | | |
|---|---|---|---|---|
| *Subject to* | $V_{to}t$ = Vi | (hi) | $W_m \geq 0$ | (g2) |
| | $A_{tot} = A_1$ | ($h_2$) | $A_m > 0$ | (g3) |
| | $V_1 = \frac{4}{3}\pi r_1^3$ | ($h_3$) | *VtZO* | (g4) |
| | $A_1 = 4\pi r_1^2$ | (h4) | *Ai>0* | (g5) |
| | Atot^Amin | (gl) | ri^O | (g6) |

where p is the density of the catalyst and $A_m$in is the required lower limit of total surface area. Step 2 of the algorithm is the specification of the extensive, intensive, assignment and input variables. They are defined below with the variable type inside parentheses:

Wtot = total weight (assignment),
$V_{to}t$ = total volume (assignment),
Atot = total surface area (assignment),
Vi = region volume (extensive),
Ai = region surface area (extensive and input),
*r* = region radius (extensive).

Constraints (hi) and (I12) specify that the total volume and total surface area be equal to those of the catalyst sphere; (I13) and (I14) define the volume and surface area of a catalyst sphere; (gi) specifies a lower bound on the total surface area; and (g2) - (ge) are the variable positivity constraints. Step 3 of the IVE algorithm is the specification of the constraint types. Constraints (hi), (h2) and (gi) - (g3) are global and the remaining are local.

Step 4 of the algorithm is the specification of the update formulae. The objective and constraints (gi) - (g3) are to remain completely unchanged across design generations.

Constraints (hi) and (h2) are the definitions of the assignment variables Vtot and $A_{tO}$t. If more than one design regions become available, the update formula for (hi) and (I12) is the sum of the particle volume and surface area of each region respectively. This concludes the input to IVE.

Monotonicity analysis indicates that this problem is *well-constrained* and *constraint-bound,* i.e. it has a finite solution with zero *DOF.* Only the positivity constraints are inactive. Ai is picked as an input variable. Note that *V* and *r* depend on A1, so only one of these three variables can be treated as an input. It is determined by backsubstitution that:

$$r_1 = \sqrt{\frac{A_{min}}{4\pi}} \text{ and } W_{tot} = \frac{4}{3}\pi\rho\left(\frac{A_{min}}{4\pi}\right)^{3/2}. \qquad (6)$$

The designer is faced with the decision of either accepting the above solution or searching for a *better* one which lies outside the boundaries of this design prototype. One way to search beyond these boundaries is through the application of IVE. In the solution prototype there are three extensive variables, no intensive variables, and two active local constraints. If constraint activity is maintained after expansion then, according to equation (3), IVE is guaranteed to increase the *DOF* of the prototype, indicating a greater possibility of finding a superior design.

Step 5 of the algorithm has been completed by choosing A1 as the critical input variable because it appears in active constraints. The initial design has one region which is expanded to two regions, the default number in step 6a, as illustrated in Figure 4. Replication of the design topology, step 6b, yields the following primitive-prototype:

*Minimize* vrtot = P •'tot

*w.r.t.* *(AIQI,V\\,V\2,A* *l,Ai2,rn,r₁₂)*

| | | | | |
|---|---|---|---|---|
| *Subject to* | Vtot = V\ i + V12 | (hi) | $V_{tot}$;≥i0 | ($g_2$) |
| | $A_m$=*An*+$A_u$ | (h2) | ^tot^¹0 | ($g_3$) |
| | $V_{11} = \frac{4}{3}\pi r_{11}^3$ | ($h_3$i) | ^n ^<3 | (g4l) |
| | $V_{12} = \frac{4}{3}\pi r_{12}^3$ | ($h_3$2) | Vl2*{3 | (g42) |
| | $A_n = 4\pi r_n^2$ | | ≥411 ^^ I3 | (g5i) |
| | $A_{12} = 4\pi r_{12}^2$ | ($h4_2$) | A12 ^ (3 | (g52) |
| | $A_{tot}$ ^ $A_{min}$ | (gi) | ni ^ (» | (g6l) |
| | | | ri2^C) | (g62) |

The notation $\_m$...*no* denotes a variable or constraint, x, in region *o,* which is derived from region *n,...,* which is derived from region *m.* Note that the global constraints are not replicated.

Instead they are modified according to their respective update formulae, as specified by step 7 of the algorithm.

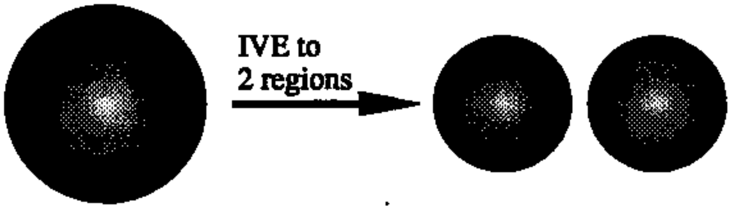



**Figure 4.** Application of IVE to the initial catalyst primitive-prototype

This primitive-prototype is also well-constrained and the resulting prototype exhibits one *DOF*9 an increase as expected. Again the positivity constraints are inactive and the remaining ones are active. Backsubstitution gives:

$$W_{tot} = \frac{4}{3}\pi\rho\left[r_{11}^3 + \left(\frac{A_{min}}{4\pi} - r_{11}^2\right)^{3/2}\right], \qquad (7)$$

and subsequent optimization produces:

$$\frac{\partial W_{tot}}{\partial r_{11}} = 0 \Rightarrow$$
$$r_{11} = r_{12} = r_{opt,2} = \sqrt{\frac{A_{min}}{8\pi}} \text{ and } W_{opt,2} = \frac{8}{3}\pi\rho\left(\frac{A_{min}}{8\pi}\right)^{3/2}. \qquad (8)$$

$r_{op}$t,2 is the value of the optimal diameters for the two-sphere design and $W_{O}$pt,2 is the corresponding optimal total weight. Comparison with the one-sphere design indicates a decrease in total weight by a factor of ^2. IVE is responsible for this improvement. Further application of IVE to both regions of the existing design proposes a four-sphere design. *Constraint dominance* is maintained and optimization gives:

$$\frac{\partial W_{tot}}{\partial r_{11}} = \frac{\partial W_{tot}}{d'12} = \frac{\partial W_{tot}}{\partial r_{21}} = 0 \Rightarrow$$
$$r_{11} = r_{12} = r_{21} = r_{22} = r_{opt,4} = \sqrt{\frac{A_{min}}{16\pi}} \text{ and } W_{opt,4} = \frac{16}{3}\pi\rho\left(\frac{A_{min}}{16\pi}\right)^{3/2}, \qquad (9)$$

**which constitutes a further improvement to the objective.**

**Responding to the trends of three consecutive design generations, we induce dominant constraint activity, where constraints (hi) - 0u) and (gi) of the initial primitive-prototype are inductively active and the positivity constraints, $(g_2)$ - (g6), are inductively inactive. IVE extended to *N* regions and optimization analysis produce the following result:**

$$r_{\mathrm{opt},N} = \sqrt{\frac{A_{\min}}{4N\pi}} \text{ and } W_{\mathrm{opt},N} = \frac{4}{3}N\pi\rho\left(\frac{A_{\min}}{4N\pi}\right)^{3/2} \Rightarrow \lim_{N\to\infty} W_{\mathrm{opt},N} = 0\,. \qquad (10)$$

The induction is successful because no constraints are violated by taking the limit of infinitely many independent spherical design regions. This solution indicates a decrease in total weight by a factor of $\sqrt{N}$, effectively proposing the design of a catalyst in powder form (many small spheres).

Other considerations, such as pressure drop, manufacturing difficulties of creating a large surface area or unpredictable catalytic behavior in very fine form may render other designs preferable, but the motivating factors for such a preference have not been included in this model. This example has served to demonstrate the power of IVE to support innovative design by finding solutions which lie beyond the design space of the initial primitive-prototype and outperform the initial design objective.

## 6. Example: Column Design

In this section an application of IVE is shown which produces a feasible design starting from an infeasible primitive-prototype. Consider the design of a clamped-clamped column under axial load (Figure 5a). The weight of the column is to be minimized subject to buckling criteria. In addition, manufacturing considerations limit the design of an individual column to a maximum acceptable weight (and effectively a maximum acceptable radius).

The primitive-prototype becomes:

*Minimize* $W_{tot}$

*w.r.t.* $(W_1, r_1, P_1)$

*Subject to* $W_{tot} = W_1$ (h1) $\qquad r_1 \ge \left[\frac{2P_1}{\pi^3 E}\right]^{1/4}$ (g1)

$W_1 = \rho\pi r_1^2 L$ (h2) $\qquad r_1 \le \left[\frac{W_{max}}{\rho\pi L}\right]^{1/2}$ (g2)

$P_1 = P_{applied}$ (h3)

The variables of this design are the radius, $r_1$, the applied load, $P_1$, which is a design input, the total weight of the design, $W_{tot}$ and the weight of a column, $W_1$. The parameters are the modulus of elasticity, $E$, the length of the column, L, the maximum allowable weight of a single column, $W_{max}$, the density of the material of construction, $\rho$, and the specified value of the applied load, $P_{applied}$.

Constraint (h1) defines the total weight; (h2) defines the weight of one column; (g1) limits the radius from buckling considerations; (g2) limits the radius from weight considerations; and (h3) specifies the desired load. Constraints (h1) and (h3) are global, while (h2), (g1) and

$(g_2)$ are local. The update formulae for both $(h_1)$ and $(h_3)$ are the sum of each column weight and axial load respectively.

Since $r$ is constrained both from above and from below, a prototype exists only if there are feasible values for the radius:

$$\left[\frac{W_{max}}{\rho\pi L}\right]^{1/2} \geq r_1 \geq \left[\frac{L^2 P_1}{\pi^3 E}\right]^{1/4} \Rightarrow P_1 \leq \frac{W_{max}^2 \pi E}{\rho^2 L^4}. \qquad (11)$$

In this primitive-prototype $P = P_{applied}$; therefore a feasible design exists only if:

$$P_{applied} \leq \frac{W_{max}^2 \pi E}{\rho^2 L^4}. \qquad (12)$$


(a) (b)

**Figure 5.** Column under axial load during the first and second design generations

Consider the situation where constraint (12) is violated, and no prototype exists. Application of IVE creates two columns and distributes the load by producing the design topology shown in Figure 5b. IVE introduces *new* design variables: each column can be modeled with an independent radius, load and material. Considering two columns of the same material, the new primitive-prototype becomes:

***Minimize*** $W_{tot}$

***w.r.t.*** $(r_{11}, r_{12}, W_{11}, W_{12}, P_{11}, P_{12})$

***Subject to***

$$\begin{array}{llll}
W_{tot} = W_{11} + W_{12} & (h_1) & r_{12} \geq \left[\frac{L^2 P_{12}}{\pi^3 E}\right]^{1/4} & (g_{12}) \\
W_{11} = \rho\pi r_{11}^2 L & (h_{21}) & r_{11} \leq \left[\frac{W_{max}}{\rho\pi L}\right]^{1/2} & (g_{21}) \\
W_{12} = \rho\pi r_{12}^2 L & (h_{22}) & r_{12} \leq \left[\frac{W_{max}}{\rho\pi L}\right]^{1/2} & (g_{22}) \\
P_{11} + P_{12} = P_{applied} & (h_3) & & \\
r_{11} \geq \left[\frac{L^2 P_{11}}{\pi^3 E}\right]^{1/4} & (g_{11}) & &
\end{array}$$

If different properties were allowed, IVE could produce different load distributions and radii; however, for the same material, at optimality the two columns have equal radii and equally divided loads. With a single material, each column carries half the load and:

' . I - ' W - & f B 4 * J ^ - **(13)**

Constraint (13) is easier to satisfy than constraint (12), so, depending on the actual values of the design parameters, FVE can produce a feasible design starting from an infeasible primitive-prototype. Repeated IVE generations produce at optimality:

$$\frac{P_{applied}}{N} \leq \frac{W_{max}^2 \pi E}{p^2 L^*}, \tag{14}$$

where *N* is the number of columns in the design. IVE will only produce a feasible prototype if at least *N* columns exist to satisfy constraint (14). The radius of each column and the total weight are:

$$r = \left[\frac{L^2 \wedge applied}{Nn^3 E}\right]^{11/4} \text{ and } W_m = pL^2 \wedge 3\&. \; E \tag{15}$$

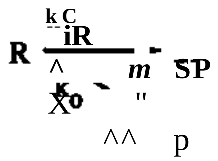
Equation (15) shows that the total weight increases with the number of columns. In conclusion, enough columns are needed to distribute the load in an acceptable fashion, as specified by constraint (14), but not more, because it works against the minimizing total weight objective, as indicated by equation (15). Application of induction produces an inferior solution in this problem. In this case IVE has innovated by producing a feasible design space which lies beyond the null design space of the initial primitive-prototype.

Note the difference between IVE and DVE. DVE would have expanded the column into a single column of different regions. IVE replicates the topology of the prototype column and relates the new columns through the total applied load.

## 7. Example: Competing Reactions in a Mixed-Reactor Network

In complex and industrially relevant applications symbolic solutions frequently cannot be found. This example demonstrates IVE in conjunction with numerical analysis.

Consider the reaction shown in Figure 6. R reacts to produce P via zero order kinetics. An unwanted side reaction produces SP via first order kinetics. The process objective is to maximize profit, as defined by equation (16).



R =Reactant

SP = Side product

P = Product

**Figure 6.** Reaction kinetics

$$Profit = \{Sales\ of\ P\} - \{Disposal\ of\ SP\} - \{Capital\ Cost\} \Rightarrow$$

$$Profit = A\ Af\ G\ C_P - B\ Af\ G\ C_{SP} - D\ V3/2 \tag{16}$$

**where**

| | |
|---|---|
| **ko = zero order rate constant [gmol/(/min)],** | ***Q* = volumetric flowrate [//min],** |
| **ki = first order rate constant [min-[1]],** | **A = sale price of TP[$/gmolP],** |
| ***C*\ = concentration of species i [gmol/fl,** | **B = disposal cost of UB [$/gmol SP],** |
| ***V* = individual reactor volume [/],** | **D = capital cost [S//$^3$^],** |
| | ***M* = time of operation [min].** |

**A more complete evaluation function would involve the cost of P, operating costs, etc.**

**The process is to be carried out by a well-mixed reactor (CSTR). The primitive-prototype takes the following form:**

*Maximize* $Profit = AMQ\ C_P - BMQ\ C_{SP} - D\ V^{3/2}$

*w.r.t.* $(C_R, C_P, C_{SP}, V, Q)$

*Subject to* $\frac{C_{R_0} - C_R}{k_0 + k_1 C_R} = \frac{V}{Q}$ (h1) $\qquad V > 0$ (g1)

$\frac{C_{P_0} - C_P}{-k_0} = \frac{V}{Q}$ (h2) $\qquad Q > 0$ (g2)

$0 \le C_R \le C_{R_0}$ (g3)

$\frac{C_{SP_0} - C_{SP}}{-k_1 C_R} = \frac{V}{Q}$ (h3) $\qquad 0 \le C_P \le C_{R_0}$ (g4)

$0 \le C_{SP} \le C_{R_0}$ (g5)

$Q = Q_0$ (h4)

**Constraints (hi) - (I13) represent the CSTR mass balances for R, P and SP respectively; (gl) and (g2) are positivity constraints; (g3) - (gs) specify the realizable concentration bounds; and (I14) sets the volumetric flowrate equal to its input value. With the exception of (I14), all constraints are local. The update formulae for IVE are: the sum of the input flowrates to each reactor for ($h_4$) and {A *M* I(Qi $C_{Pi}$) - B *M* I(Qi $C_S$Pi) - D *YV*\'¯} for the objective. Different update formulae and additional boundary conditions apply for DVE; the initial primitive-prototype is the same for both techniques, but the re-combination of the flows depends on the expansion technique.**

A specific design is defined by the following parameters:

CRo=20gmoV/, 2 = 5 //min,
$Cp_0$ =$Csp_o$ = 0gmol//, A = 5 $/gmol P,
ko =0.1 gmol/(/min), B = 0.5 $/gmol SP,
ki =0.05min-[1], D = 25,000 $//3/2,
M= 10^min(-2yrs).

At optimality it is found numerically that the CSTR volume is 153 liters, producing a profit of $ 3.6 million over a period of two years. *Better* designs can be sought by applying DVE and IVE. The corresponding design topologies are shown in Figure 7 and the results are summarized in Table 1.

**Table 1.** Comparison of the primitive prototype and the designs of DVE and IVE

| | Optimal volume, [/] | Net Profit, [106$] | P Sales, [106$] | SP Disposal, [$10^6$ $] | Capital cost, [100$] |
|---|---|---|---|---|---|
| *SingleCSTR* | 153 | 3.577 | 76.635 | 25.621 | 47.438 |
| *Serial CSTRs* (DVE) | 2@199 | 29.690 | 199.376 | 28.926 | 140.760 |
| *Parallel CSTRs* (IVE) | 2@198 | 34.583 | 197.976 | 24.112 | 139.280 |

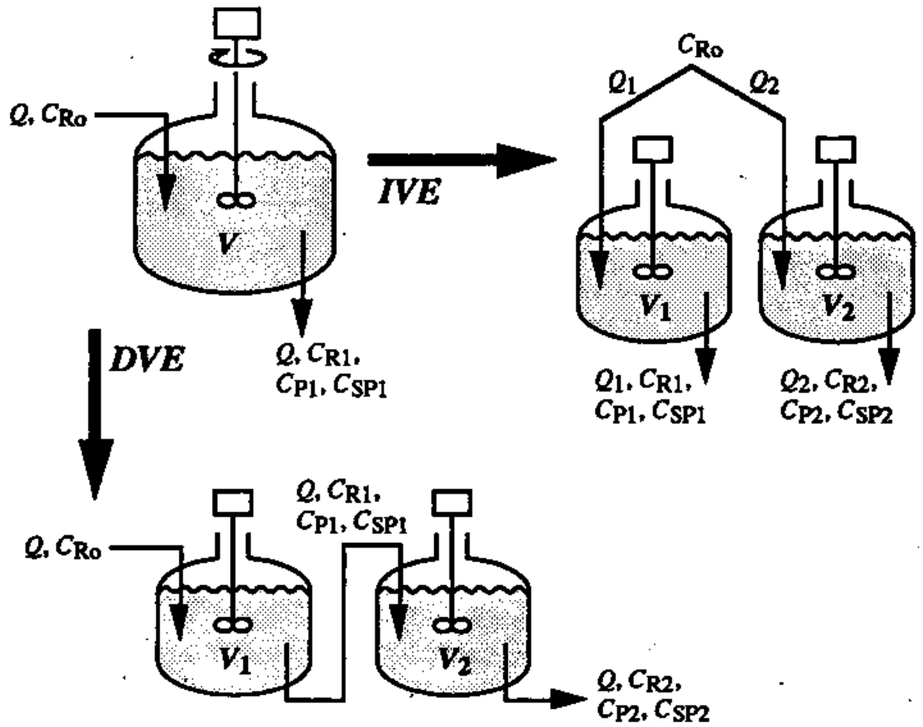



**Figure 7.** Design topologies produced by IVE and by DVE

At optimality both the serial solution, produced by DVE, and the parallel solution, produced by IVE, have two reactors of equal volumes. The parallel solution also calls for an equal split in volumetric flowrates into the two reactors. The *best* performance is exhibited by the parallel configuration, innovated through the application of IVE. The results of further application of DVE and IVE are shown in Tables 2 and 3 respectively. In both cases, at optimality the individual reactors are sized equally.

When employing a large number of reactors in series (the result of DVE, shown Table 2), the total volume of the system approaches a constant value. Inducing constraint activity, with the mass balance and flowrate equality constraints active and the positivity and concentration inequality constraints inactive, produces the limit of infinite reactors in series with total volume of about 240 liters. At this limit the capital cost vanishes, indicating that the economic model used in the process objective is no longer valid. Nevertheless the analysis shows the trend that smaller reactors in series produce a larger profit than a single reactor, because of diseconomies of scale (capital cost dependence to the 3/2 power).

Table 2. Performance of reactors in series (DVE)

| *Serial CSTRs* (DVE) | Total volume, [/] | Net Profit, | P Sales, | SP Disposal, *[!&$]* | Capital cost, *[l&$]* |
|---|---|---|---|---|---|
| 1 | 153 | 3.577 | 76.635 | 25.621 | 47.438 |
| 2 | 399 | 29.690 | 199.376 | 28.926 | 140.760 |
| 3 | 367 | 50.397 | 183597 | 31.640 | 101360 |
| 4 | 328 | 56.244 | 164.232 | 33.577 | 74.412 |
| 5 | 308 | 58.889 | 153.849 | 34.615 | 60.345 |
| 10 | 271 | 63.774 | 135.491 | 36.451 | 35.266 |
| 100 | 243 | 74.027 | 121.344 | 37.866 | 9.452 |
| 1000 | 240 | 79.102 | 120.039 | 37.996 | 2.941 |

Similarly, diseconomies of scale make the parallel operation with smaller reactors more profitable than that of a single reactor (the result of IVE, shown Table 3). Again the economic model breaks down at the limit of infinite reactors. This limit is only accessible with induction of constraint activity if at least five parallel reactors are present before the induction limit is considered because the active set of constraints changes. At this point all of R has reacted to produce P, while the mass balance of SP becomes inductively inactive and the positivity constraint on Cp becomes inductively active. In this example if we had induced constraint activity of the wrong constraint set, we would have produced an invalid solution; see Cagan and Agogino (1991b) for a discussion on the limitations of heuristic-based induction techniques.

**Table** 3. Performance of reactors in parallel (IVE)

| *Parallel CSTRs(TVE)* | **Total vol- ume,!/]** | **Net Profit, [IO^]** | **P Sales, *[l(fi$]*** | **SP Disposal, [IOH]** | **Capital cost, [lO^S]** |
|---|---|---|---|---|---|
| 1 | 153 | 3377 | 76.635 | 25.621 | 47.438 |
| 2 | 396 | 34383 | 197.976 | 24.113 | 139.280 |
| 3 | 621 | 70.812 | 310.363 | 16.333 | 223.218 |
| 4 | 841 | 108.532 | 420562 | 7.100 | 304.930 |
| 5 | 1,000 | 146.447 | 500.000 | 0 | 353553 |
| 10 | 1,000 | 250.000 | 500.000 | 0 | 250.000 |
| 100 | 1,000 | 420.943 | 500.000 | 0 | 79.057 |
| 1000 | 1,000 | 475.000 | 500.000 | 0 | 25.000 |

## 8. Discussion

These examples illustrate the fundamental features of IVE. The main idea is *parallel processing* of an input by replicating the design topology and distributing the input to the individual regions. The introduction of *new* variables supports design innovation.

In some cases, the trend towards parallelism is caused by the form of the objective function. For example, when the objective reflects diseconomies of scale then parallel designs are favored over single-region designs and IVE continues dividing the space until other features dominate the objective function. In the reactor example, the capital cost was proportional to the individual reactor volume raised to the 3/2 power, which promoted parallel processing through the diseconomies of scale.

In other cases the trend towards parallel systems generated by IVE was controlled by the design constraints. For example, in the column problem the radius limitation on each column constraint due to buckling criteria is satisfied through the repeated application of IVE.

In more complex examples there will be more interplay between other economic issues. For example, in the catalyst powder problem the costs associated with generating small particles were not included in the optimization model. As more of these features are included, direct numerical searches for performance will be preferred over symbolic methods. The reactor synthesis problem illustrates how IVE can still be used to guide the development of system structures in this more complex environment.

IVE and DVE are two main tools for expanding the design space in an optimally directed manner. We expect that other methods which are hybrids of these two will prove useful in certain problems. In addition, other expansion techniques may also be possible.

This research has not resolved the issue of how to control the application of a library of design space expansion tools, such as IVE and DVE. There may be problem characteristics and improvement trajectories which can suggest when each technique is applicable.

The IVE algorithm has not been automated. Except for the treatment of the boundary conditions, its automation would be analogous to that of DVE.

## 9. Conclusions

Input Variable Expansion, IVE, is a new formal domain independent technique for the generation of innovative designs. IVE, along with the complementary DVE, form the beginnings of a library of formal approaches for design space expansion. We expect that other complementary techniques, as well as hybrid techniques, can be invented and added to this library.

IVE has been shown to improve designs either by producing better objectives or by expanding an initially infeasible design space to create one which is feasible. The methodology has done so by employing both symbolic and numerical solutions, as needed.

IVE provides a way of formally structuring the search for alternative design concepts, be it automated or directly applied by the designer. Although symbolic analysis lends intuitive understanding to the design problem, numerical solutions are capable of addressing problems of higher complexity which may permit the application of this method to industrially relevant problems.

## 10. References

Aelion, V., J. Cagan and G. J. Powers (1991). Inducing Optimally Directed Innovative Designs from Chemical Engineering First Principles. Accepted in *Computers and Chemical Engineering.*

Androulakis, I. P. and V. Venkatasubramanian (1991). A Genetic Algorithm Framework for Process Design and Optimization. *Computers and Chemical Engineering* 15(4), 217-228.

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

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

Cagan, J. and A. M. Agogino (1991a). Dimensional Variable Expansion - A formal Approach to Design Innovation. Submitted to *Research in Engineering Design.*

Cagan, J. and A. M. Agogino (1991b). Inducing Constraint Activity in Innovative Design. In press *AI EDAM: Artificial Intelligence in Engineering, Design, Analysis and Manufacturing* 5(1).

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

Joskowitz, L. and S. Addanki (1988). From Kinematics to Shape: An Approach to Innovative Design. *Proceedings ofAAAl-88,* St. Paul, August 21-26,1:347-352.

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

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

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

Reklaitis G. V., A. Ravindran and K. M. Russel (1983). *Engineering Design Optimization.* John Wiley, New York.

Sandier, S. I. (1977). *Chemical and Engineering Thermodynamics.* John Wiley, New York.

Ulrich, K. and W. P. Seering (1988). Function Sharing in Mechanical Design. *Proceedings ofAAAI-88,* St. Paul, August 21-26,1:342-346.

Westerberg, A. W. (1989). Synthesis in Engineering Design. *Computers and Chemical Engineering* 13, 365-376.