Research in Engineering Design (1995) 7: 67–85
© 1995 Springer-Verlag London Limited

Research in
Engineering
Design

Conceptual Robustness in Simultaneous Engineering: A Formulation in Continuous Spaces

Tzyy-Shuh Chang and Allen C. Ward
Design Laboratory, Department of Mechanical Engineering and Applied Mechanics, University of Michigan, USA

Abstract. *This paper develops a robust and distributed decision-making procedure for mathematically modeling and computationally supporting simultaneous decision-making by members of an engineering team. The procedure (1) treats variations in the design posed by other members of the design team as* conceptual noise*; (2) incorporates such noise factors into conceptually robust decision-making; (3) provides preference information to other team members on the variables they control; and (4) determines whether to execute the conceptually robust decision or to wait for further design certainty. While Chang* et al. *(1994) extended Taguchi's approach to such simultaneous decision-making, this paper uses a continuous formulation and discusses the foundations of the procedure in greater detail. The method is demonstrated by a simple distributed design process for a DC motor, and the results are compared with those obtained for the same problem using sequential decision strategies in Krishnan* et al. *(1991).*

Keywords. Agents; Conceptual robustness; Concurrent engineering; Distributed concurrent engineering; Distributed decision making

1. Introduction

1.1. Motivation

Simultaneous or concurrent engineering normally refers to recent efforts in development management to increase the degree to which design tasks, especially product and manufacturing process design tasks, are conducted at the same time. The idea is not new – Henry Ford and perhaps the designers of the pyramids practiced highly concurrent design processes – but it received much attention after Clark and Fujimoto (1991) demonstrated that new model design at American motor companies took much longer than their Japanese counterparts, and attributed the difference mostly to increased concurrency in sheet metal part and die design. Early introduction of the product to the market usually implies a longer sales life and a larger market share (Smith and Reinertsen 1991), so American industry has imitated the practices that Clark and Fujimoto credited with success.

Despite the clear success and long history of simultaneous engineering, it lacks a theoretical basis. Mathematical and computing tools to support simultaneous engineering are not yet fully developed, nor do we have a mathematical model of the process. While computer science research in parallel processing should have alerted the design community that concurrent processes must often be very different from sequential ones, little attention has been paid to the logical foundation of concurrent design decision-making.

The most common, usually implicit, process model in the design literature is *hill-climbing*, in which a sequence of designs is developed, each design better than the last. Concurrent engineering efforts have focused on increasing the speed with which each successive design is passed around among the various members of the team, either by physically locating the team closer together, or by providing computational tools that accelerate communication (e.g. Bond and Ricci 1992). In this *point-based* approach, a change made by one member of the team is likely to produce changes on others, these in turn producing more changes, in a chain reaction. There is no fundamental reason for the change process ever to converge.

To minimize iterative cost, decisions about the interdependent decision variables among the components of a system (for example, the space available for an engine) or between the product and its manufacturing processes are often made by a higher-ranking manager or by one of the several involved

Correspondence and offprint requests to: Allen C. Ward, Design Laboratory, Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI 48109-2125; Tel: (313)936-0353 (email: award@dip.eecs.umich.edu).

parties in isolation (e.g. Eppinger *et al.* 1990, 1992) – that is, ignoring interdependence. This approach often results in either conflicts in the later stages of product development or lost quality, because decisions are made with insufficient data. Iteration may be required to resolve the conflicts or correct the errors.

Conversely, Ward *et al.* (1995) observe that engineers at Toyota (which has the most rapid and concurrent design process in the automobile industry) communicate about the sets of possibilities rather than changes to a single design. This communication supports increased concurrency because decision-makers can take into account the range of decisions that other members of the team may produce.

This paper moves toward a methematical model of distributed, simultaneous design decision-making based on communication about sets of possibilities. We intend that ultimately the model will satisfy several criteria. First, it should be entirely distributed, in that:

1. Different design agents can develop local design criteria which can be completely invisible to any central authority or to other design agents. A global objective function may be provided from which the agents will derive parts of their local functions, but none is required. There is no central representation of constraints.
2. Local design agents can make all design decisions; no central authority is required to mediate among local agents, establish values for interface variables, or determine in detail the sequence in which they make decisions. A central authority (or a negotiating process) need only determine which interface variables are controlled by which agents, the time at which each agent should be finished with all decisions, and the cost of delay for the entire project.
3. Local processes can proceed at different speeds, depending on the speed of the analysis and synthesis methods used by each agent.

An agent is a module (human or computational) that performs design activities based on its own knowledge and interests, interacting with others. Agents are a common idea in the distributed decision community; applications relevant to design can be found in Ahn and Crawford (1994), Cutkosky and Tenenbaum (1990), Cutkosky *et al.* (1994), Birmingham *et al.* (1993), Darr and Birmingham (1994), and Wellman (1993, 1994).

Second, the process should be *integrated*, in that the various agents take into account the most important needs of other agents, based on an efficient communication process.

Third, the model should guarantee convergence, in the sense that a solution will be found in a time appropriate to the cost of delay.

Fourth, the solution on which the model converges should be approximately optimal, in the sense that the possible difference between the solution found and the solution that would be found using a perfect, centralized, global optimization procedure is small in most situations.

Fifth, with realistic assumptions about distributed and centralized processing power, the process should be faster than a centralized one.

We have not yet established all these properties. This paper discusses an early version of the model, and its performance on a problem defined by Krishnan *et al.* (1991).

1.2. Concept

Our approach is based on *conceptual robustness*: the idea that members of a team can make simultaneous, interdependent decisions by treating their uncertainty about others' decisions as a kind of *noise*, then applying robust optimization techniques. The solutions most robust against this *conceptual noise* may not be those most appealing after the concentual noise has been eliminated; therefore, team members must consider the cost of time in deciding whether to implement the conceptually robust solution, or wait for more decisions by others. Finally, team members need to provide each other with some sense of the impact their decisions have on the rest of the design. We propose formal techniques for accomplishing these tasks.

We have done so before, in Chang *et al.* (1994), using Taguchi's parameter design methods as a starting point. In this paper, we start instead with traditional continuous optimization theory. This shift broadens the field of interest and provides insight. We develop a procedure, illustrate it on a problem taken from the distributed design literature (Krishnan *et al.* 1991), and compare our results with those produced by the sequential method originally advocated for the problem.

Section 1.3 describes generally related work. The concepts, assumptions, notation, equations, and algorithm schema for our simultaneous engineering procedure (SEP) are in Section 2. Section 3 demonstrates the SEP step by step on an illustrative example, discusses the results of different cases, and compares SEP to the sequential decision strategies (SDS) proposed by Krishnan *et al.* (1991). Section 4 discusses such issues as computational complexity and extension to constrained problems. Section 5 concludes.

1.3. Related Work

Related topics include robust design, set-based reasoning, the value of information in decision-making, timelessness, and distributed design and optimization.

This paper has adopted two important ideas advocated by Taguchi: *robustness* and search over each variable independently (Taguchi 1986, 1987; Taguchi and Clausing 1990). By extending Taguchi's method, Chang *et al.* (1994) present a procedure for simultaneous engineering while introducing the concepts of *conceptual robustness, marginal quality loss* along with *timeliness.* Wilde (1991, 1992), Otto and Antonsson (1993), and Michelena and Agogino (1991) provided methods for incorporating robustness (through not single-variable search, discussed later) into the standard optimization framework. Genetic algorithms (Goldberg, 1989), abstractions of natural evolution, provide another approach for robust design that could easily be incorporated into our framework: imitating natural selection, the robust designs "survive" after "generations".

Ward *et al.* (1995) report that the first and best practitioner of modern concurrent engineering, the Toyota Motor Company, uses a highly set-based process. Ward and Seering (1993a, 1993b) define a formal mechanism, the labeled interval calculus (LIC), for narrowing a set of possibilities by eliminating infeasible regions. However, LIC does not provide methods for making decisions within the feasible region in a distributed, simultaneous engineering configuration; the mechanisms described here do.

Bradley and Agogino (1991) address the value of information by claiming that the designer should make a selection with *negligible* expected value of perfect information (EVPI), but do not define negligibility.

Bond and Ricci (1992) state in their paper that *time* and *cost* are the factors influencing the project decisions. Ulrich *et al.* (1993) argue that apart from the design and manufacturing costs, the cost of time should be included in the decision-making process. They do not incorporate an explicit cost function for time, like the one used here.

Eppinger *et al.* (1990, 1992) and Krishnan *et al.* (1991) discuss design sequence coordination. Section 3.7 will discuss sequential design methodology (Krishnan *et al.* 1991) in detail. They address interdependence by ignoring it (making the decision in isolation), interaction, or task decoupling through an early decision by a central authority. These approaches are common in distributed design, according to Finger and Dixon (1989).

Wagner and Papalambros (1993) survey a large class of optimization work in decomposition analysis. A well-known example is the multidisciplinary design optimization techniques of Sobieszczanski-Sobieski *et al.* (Sobieszczanski-Sobieski *et al.* 1985; Sobieszczanski-Sobieski and Tulinius 1991; Consoli and Sobieszczanski 1992). Decomposition analysis is a procedure for solving well-defined optimization problems more quickly through a distributed process. We, in addition, address design activities in which such an overall optimization objective is hard to form. For instance, a power train which involves 128 variables (Wagner 1993) may have an optimization objective and require decomposition analysis to solve this optimization problem. However, designing a car may involve so many variables that it is virtually impossible to define the overall optimization objective, with parts designed in many geographic locations. Consequently, subsystems such as engine, suspension, etc., are designed by multi-functional teams separately, but not in isolation. That is, no complete overall objective function can be defined, but each team must take into account the needs of other teams in formulating its own objective. Our method, in lieu of solving the overall objective, formulates and solves cooperative design tasks. More particularly, it can be rationally used by any agent involved in a cooperative design, whether or not it is used by other agents.

2. Formulation of Simultaneous Engineering in Continuous Spaces

The performance of a design is a function of manufacturing variations and variations in the environment of use. The utility of the result of a design decision may also depend on other decisions not yet completed.

Let the *delivered cost* include the costs of manufacture, etc., as well as the *performance* cost relative to some datum design: it corresponds to the objective function to be minimized in conventional optimization. It is always possible for the designers of a subsystem to make decisions immediately, producing the best design possible *now* (though it may perform badly with some possible decisions by others). Then, we have

$$\begin{aligned}&\mathrm{E}[\text{delivered cost now}]\\&\quad = f(\text{current decisions,}\\&\qquad \text{manufacturing and environmental noises,}\\&\qquad \text{others' design decisions}),\end{aligned}$$

where $\mathrm{E}[\cdot]$ represents the expected value.

Conversely, at any time, a subsystem designer may achieve a better design by waiting for more information before making a decision. In particular, the more decisions are made first by others, the better the designer's decisions will fit the whole system. But waiting is expensive; someone has to decide first; and the other's decisions may preclude a good subsystem design. We have:

$$\begin{aligned}\text{cost of decision making}\\ &= \text{E[delivered cost } \textit{now}]\\ &\quad - \text{E[delivered cost achievable by } \textit{waiting}]\\ &\quad - \text{cost of time lost by waiting;}\end{aligned}$$

this balances the gain from waiting and the loss in time.

The delivered cost for the subsystem is not only the isolated cost directly perceivable by the subsystem designer, but also includes costs imposed on other parts of the design. These costs are hard to estimate and represent. It is easier to deal with the *marginal* cost: the increase from some datum imposed on other parts of the design by the designer's decisions. Hence:

$$\begin{aligned}\text{E[delivered cost]} &\approx \text{E[isolated cost]}\\ &\quad + \text{E[marginal costs imposed].}\end{aligned}$$

This section expands these ideas into a complete procedure.

2.1. Basic Assumptions

The design problem will be set up based on these assumptions.

1. There are only two basic kinds of variables involved in the design: physical noises, if they exist, and design variables. All the induced variables and equality constraints have been eliminated using substitution.
2. Design variables take continuous values. Chang *et al.* (1994) have demonstrated an approach to discrete-valued problems based on Taguchi's method.
3. Agents have restricted ability to understand each other's tasks, and no central authority can establish a complete overall objective function. Some variables may be identified as interface variables, of interest to more than one agent, but even for these, only a restricted amount of communication about the effect of the variable is possible. This is usually true for complex system designs (e.g. Bond and Ricci 1992).
4. The overall cost evaluation is the sum of the individual agent's cost functions, plus a constant. This implies that while actions by one agent may affect the costs of others, and there may be some overlap between the cost effects or some fixed costs that are not assigned to any agent; agents can estimate with reasonable accuracy the effect of a change in their local costs on the overall cost. This paper does not address how to set up these functions; we expect to argue in future work that both cost and profit functions are needed, with monetary units. We simply assume all the cost function outputs have the same unit – dollars, which has a common ground for value transmission from individual to individual and is additive.
5. All design processes operate over *candidate domains* of the design parameters, and all designs in these candidate domains are feasible. Ward and Seering (1993a, 1993b) have described propagation mechanisms for distributed inferences about feasibility, and at some point those methods should be integrated with these, but for the present we will assume that feasibility is enforced by methods external to this paper.
6. Design parameters are *reasonably independent*. This assumption is based on the extension of the Taguchi method, because it is well known and well studied (Otto and Antonsson 1993). Other robust design techniques may not require this assumption. We will discuss this assumption below.

We recognize that not all the design tasks satisfy these assumptions: we intend only to provide a simplified basis for early study. As discussed in Section 4, we will work at relaxing some of these assumptions in the future.

2.2. Definitions

1. There are m agents, $\mathbf{A}_i$, $i = 1, 2, 3, \ldots, m$, in a network that allows bi-directional communication between any pair of agents. In graph theory (Aho *et al.* 1983), one can view this network as a fully connected, non-directional graph by treating each agent as a node.
2. There are n design variables d_i^j spanning space $\mathbf{D}$, to be decided by the design team. We will denote any particular vector of values in $\mathbf{D}$ by $\boldsymbol{d}$, and use the same convention for other spaces.
3. Subspaces $\mathbf{D}_i$ partition $\mathbf{D}$. That is,

$$\bigcap_{i=1}^{m} \mathbf{D}_i = \varnothing, \tag{2.1a}$$

and

$$\bigcup_{i=1}^{m} \mathbf{D}_i = \mathbf{D}. \tag{2.1b}$$

The dimension of $\mathbf{D}_i$ is n_i, $\dim(\mathbf{D}_i) = n_i$, where

$$\sum_{i=1}^{m} n_i = n. \tag{2.1c}$$

Namely, for any variable d_i^j in $\mathbf{D}_i$, agent $\mathbf{A}_i$ alone will make decisions. This means individual yet interdependent decisions, rather than consensus, are expected. $\mathbf{D}_i$ can be null, in the case that $\mathbf{A}_i$ acts as an evaluator, such as a consumer. Also, there is no restriction on the partitioning – it could be object-oriented or discipline-oriented (Wagner 1993), based on the needs of the team. For example, $\mathbf{D}_i$ can be a component (engine, suspension, etc.) designer of a car; she or he can be the manufacturing engineer for the component, too.

4. $\mathbf{f}_i$ is the *isolated* cost function that $\mathbf{A}_i$ would minimize in the absence of information about the effect of her decisions on others.

5. A space $\mathbf{X}_i$ represents the distinct factors that matter in $\mathbf{A}_i$'s design. That is, $\mathbf{f}_i = \mathbf{f}_i(\boldsymbol{x}_i)$, where $\boldsymbol{x}_i \in \mathbf{X}_i$. $\mathbf{X}_i$ may include physical noises and variables controlled by other agents, as well as the variables $\mathbf{D}_i$ that agent $\mathbf{A}_i$ controls.

6. Subspace $\mathbf{C}_i$ comprises $\mathbf{A}_i$'s incoming interface variables (or conceptual noise):

$$\mathbf{C}_i = (\mathbf{X}_i \cap \mathbf{D}) \backslash \mathbf{D}_i. \tag{2.2}$$

These are the decision variables controlled by other members of the design team but important to $\mathbf{A}_i$. The definition implies $\mathbf{C}_i \cap \mathbf{D}_i = \varnothing$, and

$$\mathbf{C} \equiv \bigcup_{i=1}^{m} \mathbf{C}_i \subseteq \mathbf{D}. \tag{2.3}$$

Some design variables may not be interface variables. $\mathrm{Dim}(\mathbf{C}_i) = n_i^{\mathbf{C}}$.

7. The space of physical noises for $\mathbf{A}_i$ is

$$\mathbf{P}_i = \mathbf{X}_i \backslash (\mathbf{D}_i \cup \mathbf{C}_i). \tag{2.4}$$

$\mathbf{X}_i = \mathbf{D}_i \cup \mathbf{C}_i \cup \mathbf{P}_i$.

8. The space of $\mathbf{A}_i$'s outgoing interface variables, $\mathbf{Y}_i$, is

$$\mathbf{Y}_i = \mathbf{D}_i \cap (\mathbf{C} \backslash \mathbf{C}_i). \tag{2.5}$$

$\mathbf{Y}_i \subseteq \mathbf{D}_i$.

9. $\mathbf{E}_i$, the space of $\mathbf{A}_i$'s purely internal design variables (the variables that do not affect others' design processes), is

$$\mathbf{E}_i = \mathbf{D}_i \backslash \mathbf{Y}_i. \tag{2.6}$$

Equations (2.2) and (2.4) imply that $\mathbf{D}_i$, $\mathbf{C}_i$ and $\mathbf{P}_i$ partition $\mathbf{X}_i$. Thus, $\boldsymbol{x}_i = [\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i]$ and the cost function $\mathbf{f}_i(\boldsymbol{x}_i)$ can be written as $\mathbf{f}_i(\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i)$ where $\boldsymbol{d}_i \in \mathbf{D}_i$, $\boldsymbol{c}_i \in \mathbf{C}_i$, and $\boldsymbol{p}_i \in \mathbf{P}_i$.

2.3. Problem Formulation

Agents face uncertainty about physical noise *and* decisions to be made by other agents. For each agent, an isolated, immediate formulation of the problem is: find a d_i^{j*} for every d_i^j in $\mathbf{D}_i$, or a $\boldsymbol{d}_i^* \in \mathbf{D}_i$, such that

$$L_d(\boldsymbol{d}_i) = \int_{\mathbf{C}_i} \int_{\mathbf{P}_i | \boldsymbol{d}_i} \mathbf{f}_i(\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i) \cdot \rho(\boldsymbol{c}_i) \cdot \rho(\boldsymbol{p}_i \mid \boldsymbol{d}_i) \cdot \mathrm{d}\boldsymbol{p}_i \cdot \mathrm{d}\boldsymbol{c}_i \tag{2.7a}$$

is minimized. *Isolated* means that this objective function ignores the desires of other agents in the design team. *Immediate* means it ignores the possibility of waiting for more information. ρ is the probability density function on $\mathbf{C}_i$ and $\mathbf{P}_i$, so

$$\int_{\mathbf{C}_1} \rho(\boldsymbol{c}_i) \cdot \mathrm{d}\boldsymbol{c}_i = 1, \tag{2.7b}$$

$$\int_{\mathbf{P}_i | \boldsymbol{d}_i} \rho(\boldsymbol{p}_i \mid \boldsymbol{d}_i) \cdot \mathrm{d}\boldsymbol{p}_i = 1. \tag{2.7c}$$

$\mathbf{P}_i$ may be $\boldsymbol{d}_i$-dependent; i.e. the physical noise space may vary when the design variables are given different values. The process of specifying these probabilities is not addressed in this paper; for now, we can think of them as established by the judgment of the users, and as possibly uniform. We think that experienced engineers could be taught to estimate the accumulated probability for options with reasonable ease. For instance, to transmit a torque, one can estimate that the shaft diameter, given the materials, will not be larger than 50.8 mm (2 in) and has only 30% of the chance to go below 38.1 mm (1.5 in), based on past experience, uncertainty about load conditions, the cost of heat treatment, etc. A rough curve, ***Pr****(diameter)*, showing the accumulated probability that the shaft diameter will be smaller than *diameter*, can be sketched through these points. The derivative of this accumulated probability can serve as the probability density function in equation (2.7a).

Equation (2.7a) is the expected value of the cost function. By integrating or averaging over $\mathbf{C}_i$ and $\mathbf{P}_i$, the conceptual and physical noises, we achieve a robust design. Integration over the physical noises achieves a design that performs well over a range of physical conditions. Integration over the conceptual noise $\mathbf{C}_i$ produces design decisions by agent $\mathbf{A}_i$ that will perform well over a range of possible decisions by other members of the design team.

To further focus on the concept of conceptual robustness, we will hereafter consider only the conceptual noise $\mathbf{C}_i$. Physical noise will be implicitly included in the objective function. That is, we consider the problem

$$\text{minimize } \mathbf{F}_i(\boldsymbol{d}_i, \boldsymbol{c}_i);$$

if physical noise exists, then

$$\mathbf{F}_i(\boldsymbol{d}_i, \boldsymbol{c}_i) = \int_{\mathbf{P}_i \mid \boldsymbol{d}_i} \mathbf{f}_i(\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i) \cdot \rho(\boldsymbol{p}_i \mid \boldsymbol{d}_i) \cdot \mathrm{d}\boldsymbol{p}_i.$$

Thus, we can rewrite equation (2.7a) into

$$L_d(\boldsymbol{d}_i) = \int_{\mathbf{C}_i} \mathbf{F}_i(\boldsymbol{d}_i, \boldsymbol{c}_i) \cdot \rho(\boldsymbol{c}_i) \cdot \mathrm{d}\boldsymbol{c}_i. \qquad (2.8)$$

2.3.1. *Marginal Cost for Conceptual Noise Factors*

Equation (2.8) reflects only agent $\mathbf{A}_i$'s individual interests, without considering others' preferences on $\mathbf{Y}_i$, a subspace of $\mathbf{D}_i$. It ignores change in one agent's costs that may be induced by others' decisions. To achieve an integrated design, the decision criterion must capture other agents' costs, in addition to the costs $\mathbf{A}_i$ can directly perceive. We introduced the concept of *marginal quality loss* (Chang *et al.* 1994) to convey these preferences; here we generalize this idea to *marginal cost.*

Marginal cost enables an agent to provide feedback to neighboring agents about her preferred values for the interdependent variables controlled by those agents and the cost of deviations from those values. The agent can consider each of her conceptual noise factors individually, as if it was one of her own design decision variables. For each value of each conceptual noise factor, the agent can compute an isolated average cost, by averaging over the space of other conceptual noises and candidate values for the decision variables. The marginal cost at any value of the conceptual noise is the difference between the average cost for that value and the lowest average cost for this variable. This function provides a measure of the importance of this factor to this agent. The neighboring agent who controls this conceptual noise variable can then take the marginal cost into account.

Figure 1 illustrates this concept. Equation (2.9) gives the cost averaged over $\mathbf{A}_i$'s design space $\mathbf{D}_i$ and conceptual noise space $\mathbf{C}_i$, except c_i^j:

$$L_c(c_i^j) = \int_{\mathbf{D}_i} \int_{\mathbf{C}_i \backslash c_i^j} \mathbf{F}_i(\boldsymbol{d}_i, \boldsymbol{c}_i) \cdot \rho(\boldsymbol{d}_i) \cdot \rho(\boldsymbol{c}_i \mid c_i^j) \cdot \mathrm{d}\boldsymbol{c}_i \cdot \mathrm{d}\boldsymbol{d}_i. \qquad (2.9)$$

Again, $\rho(\boldsymbol{d}_i)$ is the probability density function of $\boldsymbol{d}_i$ and its integration over the space $\mathbf{D}_i$ equals one. Agent $\mathbf{A}_i$ will have to estimate $\rho(\boldsymbol{d}_i)$ – at worst, by making it uniform. The integration is over the candidate domains.

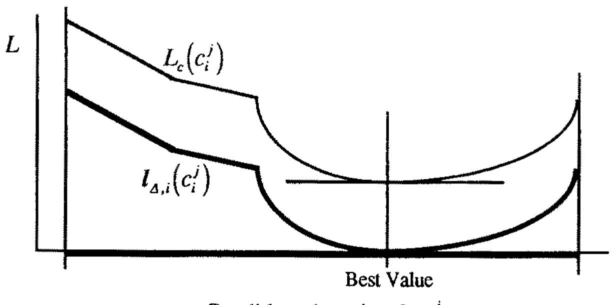

Fig. 1. Illustration of marginal cost.

The marginal cost associated with c_i^j is then:

$$l_{\Delta,i}(c_i^j) = L_c(c_i^j) - \min_{\text{candidate domain of } c_i^j} [L_c(c_i^j)]. \qquad (2.10)$$

This function will be transmitted to the agent who has control of variable c_i^j (the one who takes c_i^j as a $d_l^{\bar{j}}$ in her design space).

If we shift perspective from the conceptual noises imposed on $\mathbf{A}_i$, other agents use Eq. (2.10) to assess their marginal cost for the interface variables controlled by $\mathbf{A}_i$ – those in $\mathbf{Y}_i$. $\mathbf{A}_i$ collects these $l_{\Delta,k}(c_k^{\bar{j}})$ to compute the *total* marginal cost associated with each y_i^j as (where $y_i^j = c_k^{\bar{j}}$):

$$L_{\Delta,i}(y_i^j) = \sum_{\substack{k=1,2,\ldots,m \\ k \neq i}} l_{\Delta,k}(c_k^{\bar{j}})$$

and

$$L_{\Delta,i}(d_i^j) = \begin{cases} 0, & d_i^j \text{ not in } \mathbf{Y}_i, \\ \sum\limits_{\substack{k=1,2,\ldots,m \\ k \neq i}} l_{\Delta,k}(c_k^{\bar{j}}), & d_i^j = c_k^{\bar{j}} \text{ and } d_i^j \text{ in } \mathbf{Y}_i. \end{cases} \qquad (2.11)$$

$\mathbf{A}_i$ then adds these total marginal cost functions into her own objective function to produce an integrated design.

2.3.2. *Cost Objective and Robust Values*

We now rewrite equation (2.8) to search over one variable at a time. This yields

$$L_d(d_i^j) = \int_{\mathbf{D}_i \backslash d_i^j} \int_{\mathbf{C}_i} \mathbf{F}_i(\boldsymbol{d}_i, \boldsymbol{c}_i) \cdot \rho(\boldsymbol{d}_i \mid d_i^j) \cdot \rho(\boldsymbol{c}_i) \cdot \mathrm{d}\boldsymbol{c}_i \cdot \mathrm{d}\boldsymbol{d}_i. \qquad (2.12)$$

The outcome of Eq. (2.12), like Eq. (2.8), is subject to conceptual noise. It is also averaged over the candidate design space for the other variables. This is the approach Taguchi uses.

There are several reasons for preferring to search over one variable at a time, provided a *reasonable* degree of independence can be assured among the variables (assumption 6 in Section 2.1). Here, reasonable means that averaging over the other decisions will not produce a greatly different result than would be obtained by fixing those variables at one of the candidate values. As in Taguchi's approach, we rely on the judgment of the designer to set up the problem so that variables are reasonably independent. This will enable the direct computation of the marginal costs associated with each variable, avoiding the difficulty of allocating the cost function to the variables. That allows us to easily feed back marginal costs to other members of the design team, and to compare the costs of making a decision now with the costs of waiting. It also means that we can represent the candidate designs by a set of intervals for each variable, rather than a possibly very complex volume in the decision space. It simplifies the search process: as we will see, we need only find the zeros of functions of single variables, rather than finding level sets in the space. Averaging over the other decision variables may add an additional margin of robustness against unmodelled noises. Still, the assumption of reasonable independence is questionable, and it is possible to carry out our general approach without it by formulating the problem based on other robust design techniques. We return to these issues in Section 4 where we discuss future work.

Equation (2.12) involves only agent $\mathbf{A}_i$'s isolated costs. To incorporate others' preferences, the cost objective should include the marginal cost in the following form:

$$\hat{L}_d(d_i^j) = L_d(d_i^j) + L_{\Delta,i}(d_i^j), \qquad (2.13)$$

which combines Eqs (2.11) and (2.12). $L_d(d_i^j)$ is the isolated cost objective of $\mathbf{A}_i$; $\hat{L}_d(d_i^j)$ the integrated one.

With this cost objective, agent $\mathbf{A}_i$ can identify the robust value of d_i^j, denoted d_i^{j*}, through

$$\hat{L}_d(d_i^{j*}) = \min_{\text{candidate domain of } d_i^j} [\hat{L}_d(d_i^j)], \qquad (2.14)$$

by using any appropriate optimization technique. This value has the smallest expected cost of any value that can be selected now. Figure 2 illustrates the relations described in Eqs (2.13) and (2.14).

2.3.3. Expected Achievable Cost

The cost in Eq. (2.14) is now integrated rather than isolated, but it is still *immediate*. We need to estimate the best cost achievable if agent $\mathbf{A}_i$ waits for decisions by other agents. We will use this estimate in deciding which parts of the candidate region to retain

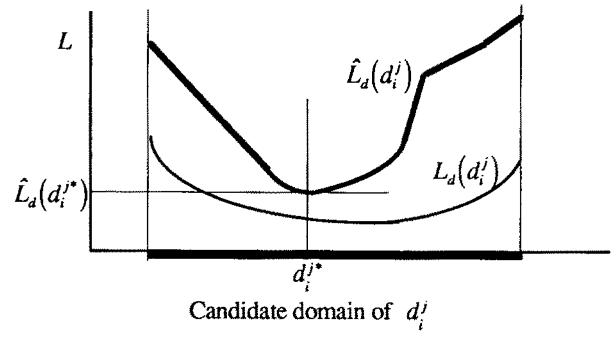

Fig. 2. Illustration of cost objective and the robust value of d_i^j.

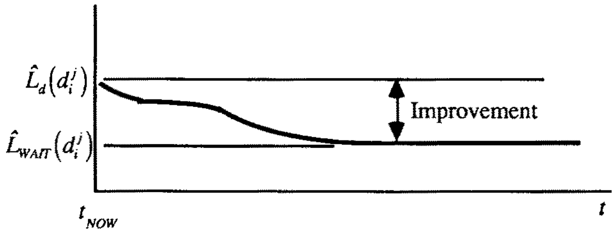

Fig. 3. Illustration of the expected achievable cost.

while waiting for more information. In Fig. 3, $\hat{L}_{WAIT}(d_i^j)$ is the asymptotic value of $\hat{L}_d(d_i^j)$ at $t \to \infty$. We claim that when used in the decision making criterion, approximations to this function should:

1. cause variable values that perform exceptionally well for some values of the interface variables to be retained until those interface values have been eliminated, or time pressure precludes waiting;
2. always cause the currently most robust value to be retained.

A variety of functions could be used to estimate $\hat{L}_{WAIT}(d_i^j)$. In this paper, we use a simple, conservative approximation: the maximum possible improvement from the immediate, integrated cost $\hat{L}_d(d_i^j)$ (Eq. 2.13). *Maximum* means that for each conceptual noise we compute the difference between the currently expected cost at each value of d_i^j and the cost achievable at that value if the conceptual noise is fixed at its most favorable value. We then take the maximum possible improvement over the entire candidate set for d_i^j, and average these possible maximum improvements. We then apply this constant to the entire cost function $\hat{L}_d(d_i^j)$, producing an offset curve that approximates the best achievable result. This procedure is *conservative* because it overestimates the possible improvement, producing a tendency to keep values in the candidate domains longer.

To convert this idea into symbolic form, we begin by considering the design variable d_i^j and the conceptual noise c_i^j. The cost, described as a function of these parameters, is:

$$L_{dc}(d_i^j, c_i^k) = \int_{\mathbf{D}_i\backslash d_i^j} \int_{\mathbf{C}_i\backslash c_i^k} \mathbf{F}_i(\boldsymbol{d}_i, \boldsymbol{c}_i) \cdot \rho(\boldsymbol{d}_i \,|\, d_i^j) \times \rho(\boldsymbol{c}_i \,|\, c_i^k) \cdot \mathrm{d}\boldsymbol{c}_i \cdot \mathrm{d}\boldsymbol{d}_i. \qquad (2.15)$$

Note that averaging $L_{dc}(d_i^j, c_i^k)$ over c_i^k yields the averaged immediate, isolated cost, $L_d(d_i^j)$, while minimizing $L_{dc}(d_i^j, c_i^k)$ over c_i^k results in the lowest cost we can hope for at any given value of d_i^j. The maximum expected improvement to be gained by keeping d_i^j in the candidate set until after c_i^k has been fixed (rather than using the robust value d_i^{j*} immediately) can then be approximated as

$$\begin{aligned}\Delta L_{d_i^j c_i^k} &= \max_{\text{candidate domain of } d_i^j} \left\{ \int_{c_i^k} L_{dc}(d_i^j, c_i^k) \cdot \rho(c_i^k) \cdot \mathrm{d}c_i^k \right. \\ &\quad \left. - \min_{\text{candidate domain of } c_i^k} [L_{dc}(d_i^j, c_i^k)] \right\} \\ &= \max_{\text{candidate domain of } d_i^j} \left\{ L_d(d_i^j) \right. \\ &\quad \left. - \min_{\text{candidate domain of } c_i^k} [L_{dc}(d_i^j, c_i^k)] \right\}. \qquad (2.16)\end{aligned}$$

$\Delta L_{d_i^j c_i^k}$ is a constant. Finally, the expected achievable cost by waiting for all the conceptual noises to be fixed before deciding on d_i^j, as a function of d_i^j, is:

$$L_{WAIT}(d_i^j) = \hat{L}_d(d_i^j) - \frac{1}{n_i^C} \sum_{k=1}^{n_i^C} \Delta L_{d_i^j c_i^k}, \qquad (2.17)$$

the immediate, integrated cost with an offset.

By using the maximum expected improvement, we meet requirements (1) and (2) mentioned in the first paragraph of this section. Values for d_i^j will be eliminated if and only if they have costs so much higher than that for the robust value that no possible improvement can justify the cost of waiting. Let ${}^2d_i^j$ produce the maximum improvement, $\Delta L_{d_i^j c_i^k}$. In the decision procedure below, it will be retained until the cost of waiting exceeds the possible gain. If ${}^1d_i^j$ results in a more robust $\hat{L}_d(d_i^j)$ than ${}^2d_i^j$, it will receive a lower L_{WAIT} than ${}^2d_i^j$; so it will be retained if ${}^2d_i^j$ is.

This approach is conservative, in the sense that it tends to overestimate the expected improvement. A more accurate but more complex approach uses the expectation of the improvement, rather than $\Delta L_{d_i^j c_i^k}$. Designers can obtain the expectation of the improvement by computing

$$\mathrm{E}\left\{L_d(d_i^j) - \min_{c_i^k} [L_{dc}(d_i^j, c_i^k)]\right\} = \int_{d_i^j} \left\{L_d(d_i^j) - \min_{c_i^k} [L_{dc}(d_i^j, c_i^k)]\right\} \cdot \rho(d_i^j) \cdot \mathrm{d}d_i^j. \qquad (2.18)$$

Here, $\rho(d_i^j)$ must be estimated by the designer. More desirable would be an improvement function based on the actual possible improvement at each value of d_i^j, and the probability of the improvement taking place. We have not found such a function that meets requirements (1) and (2).

2.3.4. *Cost of Time*

"Time is money". The problem is to establish how much money each unit of time costs. Ulrich *et al.* (1993) propose a cost of time implicit in a profit model, in which the functions of the rate of unit sale, the unit price, the unit cost, etc., are time-dependent. The model is complicated and varies from case to case. For simplicity, Chang *et al.* (1994) assume a quadratic form for the cost function of time based on Taguchi's formulation of deviation loss. The appropriate form of the cost function of time is debatable, and may vary from problem to problem. We therefore, give general criteria for the cost function of time, denoted $L_{delay,i}(t)$, without specifying a form.

We assume that agents are given individual time frames – starting and ending times – for completing their tasks by a central authority, and that an estimate is available of the total cost of delaying the project, as a function of the amount of delay. The first assumption is reasonable, because even in highly concurrent projects, some tasks must normally be performed before others. The second is reasonable given fairly standard business planning methods, for example those described in Smith and Reinertsen (1991). Consistent with, but more generally than, in our previous paper, we claim that:

1. the marginal cost of delay must increase with delay and be positive everywhere after the start;
2. the total cost of delay by any agent at $t = 0$ (the start of the agent's task) is zero;
3. the marginal cost of delay at an agent's time limit equals the cost of delaying the *entire* project.

Here the marginal cost is the derivative of the total cost of delay.

Criterion (1) is key to guarantee process convergence and is reasonable since profits decrease with increasing lead time (Smith and Reinertsen 1991). One can interpret criterion (2) as: *No loss due to delay is* *incurred if the decision is made right at the beginning of the task*; and criterion (3) as: *If the decision is not made at the due time, the whole project is delayed.*

Symbolically, we can write

$$\frac{\mathrm{d}^2 L_{delay,i}(t)}{\mathrm{d}t^2} > 0, \qquad \text{(criterion (1))} \quad (2.19a)$$

$$L_{delay,i}(\text{start time}) = 0, \qquad \text{(criterion (2))} \quad (2.19b)$$

$$\left.\frac{\mathrm{d}L_{delay,i}(t))}{\mathrm{d}t}\right|_{t=t_i} = L_{TOTAL}, \qquad \text{(criterion (3))} \quad (2.19c)$$

where t_i is the due time for $\mathbf{A}_i$ (starting from the beginning of each agent's task) and L_{TOTAL} is the cost of delaying the entire project. Criteria (2) and (3) provide initial and end conditions to help determine the cost function.

By letting t be the time *now* and $L_{TIME,i}(t)$ be the cost caused by delay (from *now* to the *next meeting*) in fixing the value of d_i^j, and assuming the design process is discretized into periods between exchanges of information,

$$L_{TIME,i}(t) = L_{delay,i}(t + t_0) - L_{delay,i}(t), \quad (2.20)$$

where t_0 is the meeting period. Figure 4 illustrates the ideas in Eqs (2.19) and (2.20).

Note that $L_{TIME,i}(t)$ is independent of d_i^j, which implies that delay on any part of $\mathbf{A}_i$'s design task costs a new iteration. Also, criterion (3) states that if $\mathbf{A}_i$ fails to decide on time, then the whole project is delayed. Imposing such a strong penalty on all delays aggressively shrinks the candidate domains. While criterion (3) seems valid, holding a delay in any of $\mathbf{A}_i$'s decisions to be equivalent to delaying $\mathbf{A}_i$'s entire task is harder to justify. A less aggressive approach might divide $\mathbf{A}_i$'s total cost of delay among the decision variables, either equally or according to the judgment of the designer.

Aggressiveness can also be reduced by imposing another constraint on the time cost function: that the marginal cost of time is zero when the task begins; thus

$$\left.\frac{\mathrm{d}L_{delay,i}(t)}{\mathrm{d}t}\right|_{t=0} = 0.$$

A more accurate but expensive assessment of the cost of delay might be achieved by assigning delay costs only to $\mathbf{A}_i$'s outgoing interface variables, and allowing the other agents to estimate the delay function for each such variable.

2.3.5. Decision Making

We are at last in a position to rewrite the statement with which we introduced this section, namely

cost of decision making

= E[delivered cost *now*]

− E[delivered cost achievable by *waiting*]

− cost of time lost by waiting,

in more precise form, by using Eqs (2.14), (2.17) and (2.20). For each d_i^j, there is a "decision-making cost function" $L_{DM}(d_i^j)$,

$$L_{DM}(d_i^j) = \hat{L}_d(d_i^{j*}) - L_{WAIT}(d_i^j) - L_{TIME,i}(t), \quad (2.21)$$

which is the estimated cost of immediately eliminating d_i^j from its candidate domain. $\mathbf{A}_i$ should keep the values of d_i^j that cause $L_{DM}(d_i^j)$ to be greater than zero in the candidate domain of d_i^j because waiting is promising. On the other hand, values causing $L_{DM}(d_i^j)$ to be less than zero should be eliminated from the candidate domain, since waiting does not look profitable. $\mathbf{A}_i$ can find the zero-crossing points by solving $L_{DM}(d_i^j) = 0$, making it easy to eliminate costly regions of d_i^j (Fig. 5). This process of narrowing the candidate domains retains the robust optima, while steadily reducing the conceptual noise. If the candidate domains of every design variable shrink by 10%, then the remaining design space is only $(9/10)^n$ the size of the original space. Since the cost of time steadily increases, the process is guaranteed to converge eventually.

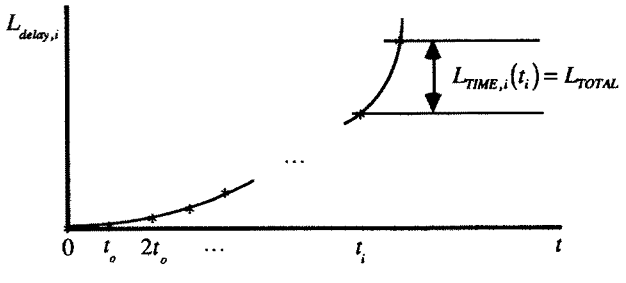

Fig. 4. Illustration of the cost function of time.

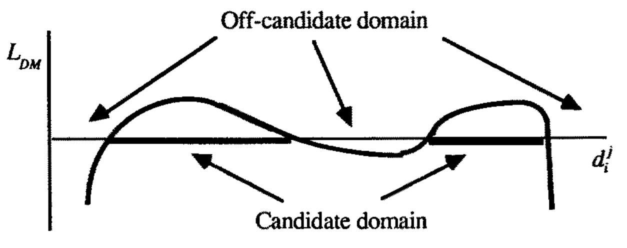

Fig. 5. Illustration of the decision-making process.

The narrowing process should iterate until

$$L_{DM}(d_i^j) \leq 0 \tag{2.22}$$

over the candidate domain of d_i^j (or the new candidate domain of d_i^j is an empty set); that is, waiting is expected to produce a higher cost than making a decision now. Then, $\mathbf{A}_i$ should pick the robust value of d_i^j, d_i^{j*}, as the final design decision. Note that d_i^{j*}, as a result of assumption (5) in Section 2.1, is a feasible solution.

2.4. Qualifications for Early Determination

Eppinger *et al.* (1990, 1992) argue that concurrent engineering implies the introduction of more coupling into the design process, making tasks harder but reducing overall iterations, and that the balance between the gain in shrinking the iterative processes and the increased effort in solving a more sophisticated problem remains indeterminate. It is therefore important to use every opportunity to determine design variable values in isolation. We here present methods for correctly fixing selected variable values without waiting for preference information.

More precisely, $\mathbf{E}_i$, $\mathbf{A}_i$'s internal design space has not been specifically addressed previously. However, the task may be greatly simplified if many of the purely internal design variables can be fixed by a purely internal process of $\mathbf{A}_i$'s without the need for information exchange with other agents. For many well-designed development processes, the dimension of $\mathbf{E}_i$ should be larger than that of $\mathbf{Y}_i$ or $\mathbf{C}_i$.

In general, if a d_i^j in $\mathbf{E}_i$ is truly (not reasonably) independent from any other variables, then we can fix it without regard to the rest of the design. Independence may be hard to determine in a noisy environment, but d_i^j in $\mathbf{E}_i$ meeting the following qualifications can be safely fixed, using the associated methods, prior to the iterative cooperation process.

(Q2.1) If d_i^j's are monotonic variables in $\mathbf{F}_i$, fix using monotonicity analysis.

(Q2.2) If d_i^j's do not cross-couple with any interface variables, fix using standard (possibly robust) optimization inside $\mathbf{A}_i$.

Proof of Q2.1

By assuming, d_i^j does not appear in any objective functions other than $\mathbf{F}_i$, and is monotonic in $\mathbf{F}_i$. Then the value of d_i^j can be easily fixed by monotonicity analysis (Papalambros and Wilde, 1988). The total cost function is the sum of the agents' cost functions plus a constant (assumption (4) in Section 2.1). Since d_i^j is monotonic in $\mathbf{F}_i$ and appears nowhere else, for any value of all other variables, the sum of the cost functions will be minimized at the same extremum of d_i^j's candidate domain. Therefore, we can fix d_i^j immediately at that extremum. □

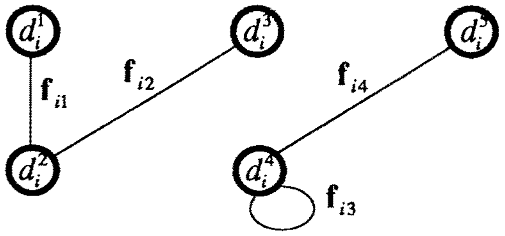

Fig. 6. Graph representation of equation (2.23).

Cross-coupling is defined using graph theory (Aho *et al.* 1983). Let each variable in $\mathbf{X}_i$ be a vertex. There is an edge between two vertices providing that there is at least one term of $\mathbf{F}_i$ containing the corresponding variables. Two variables are *cross-coupled* if they are *connected* (there exists a path between them). For example, with Eq. (2.23),

$$\mathbf{F}_i = d_i^1 \cdot d_i^2 + d_i^2 \cdot d_i^3 - d_i^4 + d_i^4 (d_i^5)^2, \tag{2.23}$$

we have the undirected graph in Fig. 6 in which $\mathbf{f}_{i1}$, $\mathbf{f}_{i2}$, $\mathbf{f}_{i3}$ and $\mathbf{f}_{i4}$ are terms of $\mathbf{F}_i$. d_i^1, d_i^2 and d_i^3 are mutually cross-coupled, while d_i^4 is cross-coupled with d_i^5. However, d_i^4 and d_i^5 do not cross-couple with d_i^1, d_i^2 or d_i^3 and vice versa. In some of the literature, $\mathbf{f}_{i3}$ and $\mathbf{f}_{i4}$, as a group, are also said to be additively separable from $\mathbf{f}_{i1}$ and $\mathbf{f}_{i2}$.

Proof of Q2.2

Let $\bar{\mathbf{E}}_i$ and $\hat{\mathbf{E}}_i$ partition $\mathbf{E}_i$, where $\bar{\mathbf{E}}_i$ is the space spanned by those design variables that are internal and do not cross-couple with any interface variables. As the total cost function is the sum of the agents' cost functions plus a constant (assumption (4) in Section 2.1), $\bar{\mathbf{E}}_i$ can be separated from $\mathbf{A}_i$'s design task such that a new agent $\bar{\mathbf{A}}_i$ will handle it with a new cost function $\bar{\mathbf{F}}_i$. $\bar{\mathbf{A}}_i$ has only internal design parameters and possibly physical noise. $\bar{\mathbf{F}}_i + \hat{\mathbf{F}}_i = \mathbf{F}_i$; $\bar{\mathbf{F}}_i$ is the part of $\mathbf{F}_i$ that does not contain any interface variables in its expression. The $d_i^j \in \bar{\mathbf{E}}_i$ can be determined earlier using optimization (if there is no physical noise), or the robust optimization techniques mentioned in Section 1.4, regardless of interface variables.

Since d_i^j appears in $\bar{\mathbf{F}}_i$ and nowhere else, for any value of all other variables, the sum of the cost functions will be minimized at the same optimizer of d_i^j. Therefore, we can fix d_i^j immediately at that optimizer. □

2.5. The Simultaneous Engineering Procedure (SEP)

The following simultaneous engineering procedure (SEP) implements conceptually robust design and summarizes Section 2.

Given a design task, for each agent $\mathbf{A}_i$, concurrently,

```
Begin
 Identify $\mathbf{D}_i$, $\mathbf{C}_i$, $\mathbf{Y}_i$, $\mathbf{E}_i$, $\mathbf{F}_i$, $t_i$ and $\mathbf{P}_i$ if applicable;
      /* Defining the problem the ith agent is encountering: design spaces, interface variables, cost function, due time, and possibly physical noises. */
 Determine some $d_i^j$'s in $\mathbf{E}_i$ through qualifications (Q2.1) and (Q2.2);
      /* Early determination process. */
 Repeat Until all $d_i^j$ are fixed:
   Transmit and receive candidate domains for $\mathbf{C}_i$, conceptual noises, and $\mathbf{Y}_i$, outgoing interface variables;
   For $j := 1$ to $n_i^C$,
     Compute $l_{\Delta,i}(c_i^j)$, marginal costs, using equations (2.9) and (2.10);
   Communicate $l_{\Delta,i}(c_i^j)$. (For $y_i^j$, receive $l_{\Delta,k}(y_i^j)$, $k = 1, 2, 3, \ldots, m$ and $k \neq i$);
   For $j := 1$ to $n_i$,
     If $d_i^j$ is not fixed,
       Begin
         Compute $L_{\Delta,i}(d_i^j)$, the total marginal cost, using (2.11);
         Compute $\hat{L}_d(d_i^j)$, the cost objective, using equations (2.12) and (2.13);
         Compute $\hat{L}_d(d_i^{j*})$, the minimum cost objective, and find $d_i^{j*}$, the robust value of $d_i^j$, using equation (2.14);
         Compute $L_{WAIT}(d_i^j)$, the expected achievable cost, using equation (2.15), (2.16) and (2.17);
         Compute $L_{TIME,i}(t)$, the cost of time, using equation (2.20);
         Compute $L_{DM}(d_i^j)$, the cost of decision making, using equation (2.21);
         If $L_{DM}(d_i^j) \leq 0$,   /* Decision made on $d_i^j$. */
           $\boldsymbol{d}_i^*[j] := d_i^{j*}$
         else
         /* Otherwise shrink the candidate domain. */
           Eliminate from the candidate domain values of $d_i^j$ that cause $L_{DM}(d_i^j) < 0$;
       end;
 endrepeat;
end.
```

On finishing the procedure, agent $\mathbf{A}_i$'s design solution is the vector $\boldsymbol{d}_i^*$.

3. Simple Illustrative Example

We now illustrate the SEP on an example introduced by Krishna *et al.* (1991) to demonstrate their sequential decision strategies for distributed design (with the units converted into the SI system). The (highly simpified) example problem is to design a DC motor by determining the decision variables listed in Table 1, maximizing the output torque (performance) and minimizing the area occupied by the stator (space) as well as the sum of the areas occupied by the steel portion of the rotor and the copper (cost of materials). Krishnan *et al.* (1991) separate the design into three tasks, accomplished by three different agents, with different objectives (as summarized in Table 2). Table 3 shows the results under the condition that designers for these tasks make their decisions in isolation (without considering others' preference). Two conflicts are observed: $dw_1^* = dw_2^* \neq dw_3^*$ and $ad_2^* \neq d_3^*$.

Krishnan *et al.* (1991) propose a sequential decision strategy (SDS) to solve this problem. They aim at minimizing the *quality loss*, which they define as the absolute value of the deviation from the optimal output obtained in isolation ($|J_i - J_i^*|$). This is different from Taguchi's quadratic form ($(J_i - J_i^*)^2$). They use the reciprocal of the magnitude of the isolated solutions, i.e. $w_i = 1/J_i^*$, to weigh the quality loss in computing the total quality loss. This quality loss function and these weightings seem to us rather arbitrary, but they do satisfy assumption (4) in Section 2.1, and we will retain them to allow direct comparison.

In this section, the design problem is solved using the SEP with the same three-agent configuration as in Krishnan *et al.* (1991); each agent is responsible for one task. In this example, there is no physical noise, $\mathbf{P}_1 = \mathbf{P}_2 = \mathbf{P}_3 = \varnothing$; adding physical noises would complicate the integrations but have no other effect.

This is an illustrative example; our purpose is to demonstrate the proposed SEP and compare it with the SDS. The scale of this problem does not convincingly need a cooperative design, and the physics and economies were simplified by Krishnan *et al.* (1991).

3.1. Identification of $\mathbf{D}_i$, $\mathbf{C}_i$, $\mathbf{Y}_i$, $\mathbf{E}_i$, $\mathbf{F}_i$ and t_i

We assign the following, with $\mathbf{D} = \{cd, dw, od, nw, ad, id, tm\}$:

For agent $\mathbf{A}_1$: (task T_1)

$\mathbf{X}_1 = \{cd, dw\}$,	$\mathbf{D}_1 = \{cd, dw\}$	
$\mathbf{C}_1 = \varnothing$,	$\mathbf{Y}_1 = \{dw\}$,	$\mathbf{E}_1 = \{cd\}$,
$t_1 = 10$ days,	$\mathbf{F}_1(\boldsymbol{d}_1) = w_1\lvert J - J_1^*\rvert$.	

Table 1. Design variables for a d.c. motor (Krishnan *et al.* 1991).

Decision variables	*Symbol*	*Bounds*
Armature diameter	*ad*	$0.254 \leq ad \leq 0.3048$ (meters)
Motor inner diameter	*id*	$0.002\,54 \leq id \leq 0.0762$ (meters)
Motor outer diameter	*od*	$0.508 \leq od \leq 0.6096$ (meters)
Diameter of windings	*dw*	$0.000\,254 \leq dw \leq 0.005\,08$ (meters)
Current density	*cd*	$155 \leq cd \leq 77\,500$ (amp/meter^2)
Turns of armature windings	*nw*	$1 \leq nw \leq 1500$
Thickness of magnet used	*tm*	$0.001\,27 \leq tm \leq 0.0254$ (meters)

Table 2. Objectives as functions of variables (Krishnan *et al.* 1991).

Task	*Task description*	*Local objective functions to be minimized*
T_1	Maximize output torque	$J_1 = -1.57cd \cdot dw^2$
T_2	Minimize space	$J_2 = 0.785(od^2 - ad^2) - 0.26nw \cdot dw^2$
T_3	Minimize material cost	$J_3 = 0.785(ad^2 - id^2) - 2.1ad \cdot tm + 0.785dw^2$

Table 3. Isolated decisions (Krishnan *et al.* 1991).

Task	*Isolated decisions*
T_1	$cd_1^* = 77\,500$; $dw_1^* = 0.005\,08$; $J_1^* = -3.14$
T_2	$ad_2^* = 0.3048$; $od_2^* = 0.508$; $dw_2^* = 0.005\,08$; $nw_2^* = 1500$; $J_2^* = 185.36$
T_3	$ad_3^* = 0.254$; $id_3^* = 0.0762$; $dw_3^* = 0.000\,254$; $tm_3^* = 0.0254$; $J_3^* = 50.4351$

For agent $\mathbf{A}_2$: (task T_2)

$$\mathbf{X}_2 = \{od, nw, ad, dw\},\quad \mathbf{D}_2 = \{od, nw, ad\},$$
$$\mathbf{C}_2 = \{dw\},\quad \mathbf{Y} = \{ad\},\quad \mathbf{E}_2 = \{od, nw\},$$
$$t_2 = 10 \text{ days},\quad \mathbf{F}_2(\boldsymbol{d}_2, \boldsymbol{c}_2) = w_2|J_2 - J_2^*|.$$

For agent $\mathbf{A}_3$: (task T_3)

$$\mathbf{X}_3 = \{id, tm, ad, dw\},\quad \mathbf{D}_3 = \{id, tm\},$$
$$\mathbf{C}_3 = \{ad, dw\},\quad \mathbf{Y}_3 = \varnothing,\quad \mathbf{E}_3 = \{id, tm\},$$
$$t_3 = 10 \text{ days},\quad \mathbf{F}_3(\boldsymbol{d}_3, \boldsymbol{c}_3) = w_3|J_3 - J_3^*|.$$

The cost functions are in the quality loss form defined by Krishnan *et al.* (1991). In $\mathbf{F}_i$, both w_i and J_i^* are constants which do not affect the isolated optimization results. However, as the cost evaluation may be communicated to other agents, they are included for proper value scaling. Assigning market values to time, and finding realistic costs in monetary units are critical tasks in cooperative design, but we have not addressed them in this paper: we therefore use the weights proposed by Krishnan *et al.* (1991). The design team will meet daily ($t_0 = 1$ day); agents have the same starting time; and the cost of delaying the entire project, L_{TOTAL}, is the sum of the weighted, isolated optimal results, $w_1J_1^* + w_2J_2^* + w_3J_3^*$.

3.2. Determination of d_i^j's in $\mathbf{E}_i$ by Checking Qualifications (Q2.1) and (Q2.2)

For $\mathbf{A}_1$, *cd* is the only internal variable. It satisfies qualification (Q2.1), because it causes $\mathbf{F}_1(\boldsymbol{d}_1)$ to decrease monotonically. Thus $\mathbf{A}_1$ will fix *cd* at its upper bound; $cd^* = 77\,500$ amp/meter^2.

Regarding *od* and *nw*, $\mathbf{A}_2$ will find that both satisfy (Q2.1) (are monotonic variables in $\mathbf{F}_2(\boldsymbol{d}_2, \boldsymbol{c}_2)$) while the former also satisfies (Q2.2) (does not cross-couple *ad* or *dw*). Her decisions on these two variables will be $od^* = 0.508$ meters (the lower bound) and $nw^* = 1500$ turns (the upper bound).

For $\mathbf{A}_3$, *id* satisfies (Q2.1) as well as (Q2.2); it causes $\mathbf{F}_3(\boldsymbol{d}_3, \boldsymbol{c}_3)$ to decrease monotonically and does not cross-couple *ad* or *dw*. *tm* satisfies (Q2.1); it causes $\mathbf{F}_3(\boldsymbol{d}_3, \boldsymbol{c}_3)$ to decrease monotonically. They should be determined as 0.0762 meters and 0.0254 meters (their upper bounds), respectively.

Thus, only *dw* and *ad* remain undetermined.

3.3. Iterative Narrowing of the Candidate Domains

3.3.1. Transmission and Reception of Candidate Domains for $\mathbf{C}_i$ *and* $\mathbf{Y}_i$

At this step, agents communicate about the candidate domains of the interface variables, *dw* and *ad*. The messages are:

From $\mathbf{A}_2 \rightarrow \mathbf{A}_3$:

$$0.254 \leq ad \leq 0.3048 \text{ (meters)};$$

From $\mathbf{A}_1 \rightarrow \mathbf{A}_2, \mathbf{A}_3$:

$$0.000254 \leq dw \leq 0.00508 \text{ (meters)}.$$

We assume that all the values in the candidate domain of a design variable are equally likely to be selected so the communication does not include probability functions.

3.3.2. Computation and Communication of $l_{\Delta,i}(c_i^j)$ Using Eqs (2.9) and (2.10)

Here each agent computes the marginal cost for each of her conceptual noise factors (incoming interface variables):

$\mathbf{A}_1$: N/A.

$$\mathbf{A}_2\text{: } L_c(dw) = \frac{1}{\|ad\|}\int_{ad=0.254}^{0.3048} \mathbf{F}_2(\boldsymbol{d}_2, \boldsymbol{c}_2)|_{od^*,nw^*}\cdot \mathrm{d}\, ad$$
$$= -3261.2279dw^2 + 0.1802,$$

where $\|\cdot\|$ denotes the size of the variable's candidate domain, and $\min_{dw}[L_c(dw)] = 0.0960$,

$$\Rightarrow \quad l_{\Delta,2}(dw) = L_c(dw) - \min_{dw}[L_c(dw)]$$
$$= -3261.2279dw^2 + 0.0842.$$

$$\mathbf{A}_3\text{: } L_c(dw) = \frac{1}{\|ad\|}\int_{ad=0.254}^{0.3048} \mathbf{F}_3(\boldsymbol{d}_3, \boldsymbol{c}_3)|_{id^*,tm^*}\cdot \mathrm{d}\, ad$$
$$= 24.1251dw^2 + 0.2904,$$

and $\min_{dw}[L_c(dw)] = 0.2904$,

$$\Rightarrow \quad l_{\Delta,3}(dw) = L_c(dw) - \min_{dw}[L_c(dw)]$$
$$= 24.1251dw^2;$$

$$L_c(ad) = \frac{1}{\|dw\|}\int_{dw=0.000254}^{0.00508} \mathbf{F}_3(\boldsymbol{d}_3, \boldsymbol{c}_3)|_{id^*,tm^*}\cdot \mathrm{d}dw$$
$$= 24.1251ad^2 - 1.6393ad - 1.1399,$$

and $\min_{ad}[L_c(ad)] = 0.0002$,

$$\Rightarrow \quad l_{\Delta,3}(ad) = L_c(ad) - \min_{ad}[L_c(ad)]$$
$$= 24.1251ad^2 - 1.6393ad - 1.1401.$$

After the above computations, these marginal cost functions are transmitted to associated agents.

3.3.3. Computation of $L_{\Delta,i}(d_i^j)$ Using Eq. (2.11)

After receiving the marginal cost functions, each agent computes the total marginal cost function for her outgoing interface variables.

$$\mathbf{A}_1\text{: } L_{\Delta,1}(dw) = l_{\Delta,2}(dw) + l_{\Delta,3}(dw)$$
$$= -3237.1028dw^2 + 0.0842.$$

$$\mathbf{A}_2\text{: } L_{\Delta,2}(ad) = l_{\Delta,3}(ad)$$
$$= 24.1251ad^2 - 1.6393ad - 1.1401.$$

$\mathbf{A}_3$: N/A.

3.3.4. Computation of $\hat{L}_d(d_i^j)$ Using Eqs (2.12) and (2.13)

With the total marginal cost and equation (2.12), each agent computes the integrated cost function for each of the design variables that she controls.

$$\mathbf{A}_1\text{: } L_d(dw) = \mathbf{F}_1(\boldsymbol{d}_1)|_{cd^*}$$
$$= -38\,750.0775dw^2 + 1,$$
$$\Rightarrow \quad \hat{L}_d(dw) = L_d(dw) + L_{\Delta,1}(dw)$$
$$= -41\,987.1803dw^2 + 1.0842.$$

$$\mathbf{A}_2\text{: } L_d(ad) = \frac{1}{\|dw\|}\int_{dw=0.000254}^{0.00508} \mathbf{F}_2(\boldsymbol{d}_2, \boldsymbol{c}_2)|_{od^*,nw^*}\cdot \mathrm{d}\, dw$$
$$= -6.5643ad^2 + 0.6645,$$
$$\Rightarrow \quad \hat{L}_d(ad) = L_d(ad) + L_{\Delta,2}(ad)$$
$$= 17.5609ad^2 - 1.6393ad - 0.4756.$$

$\mathbf{A}_3$: N/A.

3.3.5. Computation of $\hat{L}_d(d_i^{j*})$ and Find d_i^{j*} Using Eq. (2.14)

In this step, agents find the robust values for the design variables – those that minimize the expected cost functions.

$\mathbf{A}_1$: $\hat{L}_d(dw^*) = 0.0006$, $\quad dw^* = 0.005\,08$.

$\mathbf{A}_2$: $\hat{L}_d(ad^*) = 0.2410$, $\quad ad^* = 0.254$.

$\mathbf{A}_3$: N/A.

3.3.6. Computation of $L_{WAIT}(d_i^j)$ Using Eqs (2.15), (2.16) and (2.17)

Agents now estimate the expected achievable cost.

$\mathbf{A}_1$: There is no conceptual noise ($\mathbf{C}_1 = \varnothing$). Therefore, the decision should be $dw^* = 0.005\,08$, because waiting is unnecessary.

$$\mathbf{A}_2\text{: } L_{dc}(ad, dw) = \mathbf{F}_2(\mathbf{d}_2, \mathbf{c}_2)|_{od^*, nw^*}$$
$$= 0.6940 - 6.5643ad^2 - 3261.2279dw^2,$$

$$\Delta L_{ad,dw} = \max_{ad}\left\{L_d(ad) - \min_{dw}[L_{dc}(ad, dw)]\right\}$$
$$= 0.0546,$$

$$\Rightarrow \quad L_{WAIT}(ad) = \hat{L}_d(ad) - \Delta L_{ad,dw}$$
$$= 17.5609ad^2 - 1.6393ad - 0.5302.$$

(Note that $\mathbf{A}_1$ has made a decision on *dw*. With prompt communication, $\mathbf{A}_2$ can make a deterministic decision on *ad* without computing $L_{WAIT}(ad)$ and $L_{TIME,i}(t)$. However, in order to illustrate the process, we assume that this piece of information is not communicated until the next meeting.)

$\mathbf{A}_3$: N/A.

When there is no conceptual noise (as for $\mathbf{A}_1$), then the expected available cost is the immediate, integrated cost objective.

3.3.7. Computation of $L_{TIME,i}(t)$ Using Eq. (2.20)
We will use the quadratic form, $L_{delay,i}(t) = k_t \cdot t^2$, suggested in Chang *et al.* (1994), with $t_0 = 1$ day, $t_i = 10$ days, $L_{TOTAL} = 3$. From equation (2.19c),

$$k_t = \frac{L_{TOTAL}}{2t_i} = \frac{3}{2 \cdot 10}.$$

The quadratic form satisfies the criteria in equations (2.19). For the first iteration, equation (2.20) yields

$$L_{TIME,1}(1) = L_{TIME,2}(1) = L_{TIME,3}(1) = 0.15.$$

3.3.8. Computation of $L_{DM}(d_i^j)$ Using Eq. (2.21)
Combining the results in Sections 3.3.5, 3.3.6 and 3.3.7, we have the cost of immediate decision-making:

$\mathbf{A}_1$: N/A.

$\mathbf{A}_2$: $L_{DM}(ad) = -17.5609ad^2 + 1.6393ad + 0.6212.$

$\mathbf{A}_3$: N/A.

3.3.9. Decision Making
Figure 7 shows that $L_{DM}(ad)$ is negative within its candidate domains. Thus, the iteration stops here; the answer found for *ad* is 0.254 meters.

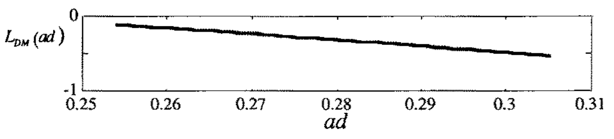

Fig. 7. L_{DM} over the candidate domain.

The solution found using the SEP process can be summarized as

$ad^* = 0.254$ meters,	$id^* = 0.0762$ meters,
$od^* = 0.508$ meters,	$dw^* = 0.005\,08$ meters,
$cd^* = 775\,00$ amp/meter^2,	$nw^* = 1500$ turns,
$tm = 0.0254$ meters.	

It duplicates that of Krishnan *et al.* (1991). Readers can easily verify that this is the global optimizer for the DC motor design. Notice that the global optimum happens at values of the isolated optimal decisions ($ad^* = ad_3^*$ and $dw^* = dw_1^*$).

3.4. Different Weightings

This section explores what happens when the global optimum lies off all the isolated optimal decisions. To do so, the weighting is switched to $w_1 = w_2 = 1$ and $w_3 = \frac{1517}{1727}$ (left in fraction form because this problem is very sensitive to the number). The SEP will find the global optimum at $ad^* = 0.2794$ meters the sequential strategy will continue to return the old value, 0.254.

3.4.1. Result of Sequential Decision Strategies (SDS)
Using the SDS procedure proposed by Krishnan *et al.* (1991), the optimal sequence is either $T_1 \to T_2 \to T_3$ or $T_1 \to T_3 \to T_2$. The solution is the same as in Section 3.3, except that *ad* can be either 0.254 or 0.3048. The resulting total quality loss is 30.3713. SDS fails to find the global optimum.

3.4.2. Results of the SEP
Following the SEP, we have

$\mathbf{A}_1$: $dw^* = 0.005\,08$ (*no conceptual noise*).

$\mathbf{A}_2$: $L_{DM}(ad) = -147.9548ad^2 + 82.6772ad - 13.1602,$
$ad^* = 0.2794.$

$\mathbf{A}_3$: N/A.

Again, Fig. 8 shows that $L_{DM}(ad)$ is negative within its candidate domain and the answer found for *ad* is 0.2794 meters.

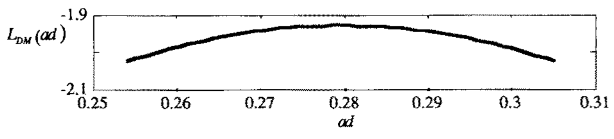

Fig. 8. L_{DM} over the candidate domain.

The summarized solution,

$$ad^* = 0.2794 \text{ meters}, \quad id^* = 0.0762 \text{ meters},$$
$$od^* = 0.508 \text{ meters}, \quad dw^* = 0.00508 \text{ meters},$$
$$cd^* = 77500 \text{ amp/meter}^2, \quad nw^* = 1500 \text{ turns},$$
$$tm^* = 0.0254 \text{ meters},$$

coincides with the global optimum.

3.5. Ignorance of the Qualifications for Early Determination

In Section 3.2, we fixed the values of *cd*, *od*, *nw*, *id* and *tm* without iteration, using qualifications (Q2.1) and (Q2.2). What if we did not do so? As an example, $L_c(dw)$ for agent $\mathbf{A}_2$ would become

$$L_c(dw) = \frac{1}{\|od\| \cdot \|nw\| \cdot \|ad\|} \int_{ad=0.254}^{0.3048} \int_{nw=1}^{1500} \int_{od=0.508}^{0.6096} \times \mathbf{F}_2(\boldsymbol{d}_2, \boldsymbol{c}_2) \cdot \mathrm{d}\, od \cdot \mathrm{d}\, nw \cdot \mathrm{d}\, ad,$$

which requires much more computation than the one in Section 3.3.2.

Even worse, this more complex process produces an inferior solution. Using the weighting function set in Section 3.4, following the SEP without the qualification check changes the result such that $ad^* = 0.254$ meters. This is because averaging over values of *tm* from its optimal value to the lower bound weakens the advantage of a large *ad* (or the influence of the term $-2.1 ad \cdot tm$). In other words, this ad^* (0.254) is a robust response to the uncertainty of *tm*, but this robustness is unnecessary because *tm* can be fixed earlier.

Furthermore, higher uncertainty usually requires more iterations to converge.

Fortunately, in well-decomposed design problems, we can hope that a majority of variables can be fixed by isolated processes. The criteria for doing so are an important feature of the SEP.

3.6. The Influence of the Cost of Time

If we set $L_{TIME,i}(t) = 0$ (no time pressure), the result of the first iteration in Section 3.4.2 fixes *dw* only. An additional iteration is needed, but the final solution is the same, showing the cost of time contributes to rapid convergence. In some cases, in which time pressure is very strong and the information is quite vague, the solutions may be quite different with and without time pressure.

3.7. Comparison with Sequential Decision Strategies

The simultaneous engineering procedure (SEP) appears superior to the sequential decision strategy (SDS) in this respect: the SEP is not restricted to isolated, local results, and thus may achieve or approximate the true optimum for a wider variety of problem types.

SDS works well when the global optimum coincidentally happens at the values where isolated optima lie. However, it fails to locate the global optimum in the case depicted in Section 3.4. Actually, for this particular case, the ad^* found using the optimum sequence in SDS is located as far as possible from where the globally optimal value locates.

Conversely, for this case SEP can quickly converge to a good solution. Still, the result obtained in Section 3.4.2 does not imply that the integrated solution found using SEP will duplicate the global optimum in every case: there is a price to be paid for concurrency. At best, we hope eventually to be able to estimate the expected deviation from the true optimum as a function of the level of time pressure.

Despite the performance difference, there are interesting similarities between the ideas proposed by Krishnan *et al.* (1991) and ours. Their idea of using *exclusive groups* is close to our idea of design space partitioning, while our marginal cost is somewhat akin to their partial quality loss.

However, Krishnan *et al.* (1991) do not allow their exclusive groups to communicate, but rely on communication by the groups with a central authority that sequences their decisions. This approach restricts their overall result to some combination of the results achievable in isolation. We do not require sequencing or much central authority, but do allow our agents to communicate.

Our marginal cost provides feedback to integrate the agents' decisions, considering the influence of the yet-to-be fixed variables. Their partial quality loss does neither, but is a substitute for a global cost function.

To conclude, the fundamental distinction is that they make decisions early in isolation, then adjust the decision sequence to minimize the cost of isolation, while we leave the uncertain decisions open until we have sufficient information to fix them.

4. Discussion and Future Work

We have defined a procedure, and demonstrated it on a simple problem. However, the work raises as many questions as it answers. We first discuss some theoretical issues; then a practical one.

4.1. Foundations

The SEP seems plausible to us, but there is no formal connection between our assumptions and the procedure. We would like to axiomatize the procedure. The treatment of game theory by Van Neumann and Morgenstern (1964) is a suggestive example, because SEP is a form of cooperative game.

We assume that individual agents have their own cost evaluations and the results are all in a common, additive unit with an external reference (e.g. dollars). This assumption may be relaxed somewhat if the agents act as buyers and sellers, as in Wellman (1994), Vasseur *et al.* (1993), and Cook (1992). The goods in this case would be the possibilities (or options) of the design variables and the price bidding would show the preferences of the buyers (interested agents). The "trade" among agents represents the communication and the "bidding strategies" (decisions) are based on utility functions, no matter whether they are additive or not. Consumers could also be agents in the design group.

4.2. Computational Complexity vs Optimality

Most computations in the SEP are integrations. Integration can usually be done numerically, if not analytically, but may be slow. Even without analytical equations, empirical data can be used to find the solution by curve-fitting, or by replacing the integrals with summations. The procedure can therefore easily be extended to the mixed discrete–continuous case.

The other computational problem is to find the candidate domain. We solve for one variable at a time – a line search. The example uses graphical methods, but the solution can be obtained formally by solving $L_{DM}(d_i^j) = 0$, finding the zero-crossing points.

However, we have yet to define a measure of the computational cost of the SEP. Its complexity, as a function of the size of the problem, depends on the amount of time pressure: there is a tradeoff between how close the answer comes to the global optimum, and the time required to find it (Eppinger *et al.* 1990, 1992). We would like to derive an analytical formula for this tradeoff, but it seems unlikely, since no centralized procedure can guarantee a global optimum in finite time. Perhaps some probabilistic result, or one dependent on the problem, is possible.

Without such a theory, or experiments, we cannot even be confident that our distributed process is faster than a centralized, standard optimization process. It may be useful even if slower, simply because it does not require a single model to be formulated. But our guess is that it can be much faster: we need a way to test the guess.

4.3. Variable Independence and Constrained Problems

We assumed feasibility within the candidate domains and the *reasonable independence* of variables, so that we can use line search. We have not developed a precise measure of reasonable independence, nor do we know how common problems can be reformulated to provide reasonable independence, although users of Taguchi's methods seem to incorporate this assumption with fairly good affects. It seems possible that by using appropriate combinations of variables, adding variables, eliminating equality constraints by substitution, and replacing inequality constraints with penalty functions (see Chapter 9 in Bazaraa 1993), or perhaps the quadratic penalty functions used by Taguchi), most problems can be recast to involve only reasonably independent, unconstrained decision variables. Formalizing and proving this conjecture is a research task.

Another approach is to interleave constraint reasoning based on intervals, as in Ward and Seering (1993a, 1993b), with SEP. In this case feasibility and candidacy will be examined in parallel. This combination allows the constraints to be built-up, instead of specified firmly in the beginning – an advantage, because designers often do not initially fully understand how, when, and where to apply what constraints.

A third approach is to use constraint checking only during minimization, at Eqs (2.7), (2.10) and (2.14). The presence of noise complicates the meaning of constraints, since a solution may violate the constraints for some circumstances but not for others. This problem can be addressed by using a threshold probability of satisfying the constraints (Siddall 1983; Otto and Antonsson 1993). That is, the constraints in the problem

$$\left.\begin{array}{ll}\text{minimize} & \mathbf{F}_i(\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i) \\ \text{subject to} & \mathbf{g}_i^{\bar{j}}(\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i) \le 0\end{array}\right\} \tag{4.1}$$

can be replaced by probabilistic constraints:

$$\min_{\bar{j}}[\boldsymbol{Pr}(\mathbf{g}_i^{\bar{j}}(\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i) \le 0)] \ge \delta.$$

Here $\bar{j}$ indexes the constraints and δ is the acceptable probability threshold of meeting constraints. For instance, suppose $\mathbf{A}_i$ has constraints as in Eq. (4.1): she would do the constraint checking using

$$\min_{\bar{j}}\left[\underset{\substack{\mathbf{D}_i\backslash d_i^j\\ \mathbf{C}_i, \mathbf{P}_i}}{\boldsymbol{Pr}}\left(\mathbf{g}_i^{\bar{j}}(\boldsymbol{d}_i, \boldsymbol{c}_i, \boldsymbol{p}_i)\right)\right] \ge \delta \tag{4.2}$$

in conjunction with the minimization of (2.14). For the operations associated with Eqs (2.7) and (2.10), the examined spaces in Eq. (4.2) would be changed.

Probabilistic constraint checking is hard to incorporate into qualifications (Q2.1) and (Q2.2) for early value determination, because the bounds of domains are not always available. Nevertheless, agents should try to reduce the number of parameters by conducting monotonicity analysis.

Constraint checking would allow relaxation of the feasibility assumption 5 in Section 2.1. The candidate domains could then range over *all* rational values to the design variables at the beginning of the process.

4.4. Practical Utility

The example problem is too simple to provide real feedback on the utility of the SEP for large-scale practical engineering problems. The great advantage of the SEP is that it does not require either a centralized utility function, or compatibility in speed or level of detail in the models used by the design agents. It is possible, for example, for one agent to explore a wide variety of solutions using a fast analytical model, achieving a highly optimized subsystem, while another uses a complex finite element code to achieve a merely good solution based on a limited number of runs. To explore these advantages more fully, we will need to test it on real problems. We are currently working with Ford and Whirlpool on industrial tests.

5. Conclusion

In this article we have extended the conceptual robustness approach to simultaneous engineering from discrete levels to continuous spaces. With the assumptions made, each designer in the team can identify both physical and conceptual noise. She will make robust decisions based on a criterion which comprises expected quality loss and timeliness. The criterion is easily implemented in many circumstances, and is based on plausible considerations, but we have not formalized it. The design process must converge because of a satisfactory result, or the time pressure, or both to approximately minimize the overall cost (where we have not formalized "approximately").

The qualifications for early determination contribute significantly to problem simplification, and establish a partial ordering of tasks, increasing both speed and solution quality.

The example illustrates an advantage of the proposed simultaneous engineering procedure in comparison with sequential designs, and the contribution of the qualifications for early determination and the cost of time.

The proposed procedure is still subject to limitations embedded in the assumptions. First, the functions used to compute induced design variables (or performance variables) from design variables have to be communicated along with the domains of the design variables. The substitution of these functions into the cost objective functions may be cumbersome.

Second, the global reference of the cost evaluation is hard to define for a large-scale design task. Believing that utility theory does not provide effective evaluation among agents because it is inherently individual, we envision that *cost* and *revenue* concepts, as in market economics, will play a major role in distributed design plans.

Third, there are a variety of possible strategies for dealing with constraints, and we do not know how to compare them yet.

Innovative concepts in this paper are closely related to both "distributed design" (Finger and Dixon, 1989) and distributed optimization. In distributed design as defined by Finger and Dixon, interdependent variables are fixed through a hierarchical decision structure or in isolation. Conversely, we advocate parallel decision making with peer-to-peer communication to integrate the design. Distributed optimization work normally addresses large optimal design problems that are decomposed for easier solution. We envision the reverse process, in which local problems are partially integrated through an efficient communication process. We will further explore these concepts in real engineering applications as well as through analytical studies.

Acknowledgments

This research is funded by the National Science Foundation under grant No. DDM 9300376, and by the Office of Naval Research under grant N00014-91-J-1918. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the sponsors. We are also grateful to the anonymous reviewers for their thorough reviews and valuable comments.

Table of Nomenclature

$\mathbf{A}_i$	the ith design agent
$\mathbf{C}_i$	the space of conceptual noise factors of $\mathbf{A}_i$
c_i	a vector in $\mathbf{C}_i$
c_i^j	the jth conceptual noise factor or one-dimensional subspace in $\mathbf{C}_i$
$\mathbf{D}$	the space containing all the design variables

$\mathbf{D}_i$	the design space controlled by $\mathbf{A}_i$
$\boldsymbol{d}_i$	a vector in $\mathbf{D}_i$
$\boldsymbol{d}_i^*$	the solution of $\mathbf{A}_i$'s design
d_i^j	the jth design variable or one-dimensional subspace in $\mathbf{D}_i$
$d_i^{j^*}$	the value of d_i^j that minimizes the cost $\hat{L}_d(d_i^j)$ (the jth element of $\boldsymbol{d}_i^*$)
$\mathbf{E}_i$	the space of $\mathbf{A}_i$'s purely internal design variables
$\mathbf{f}_i$	the cost function of $\mathbf{A}_i$ in which physical noise is explicit
$\mathbf{F}_i$	the cost function of $\mathbf{A}_i$; $\mathbf{f}_i$ integrated over the physical noise
$\mathbf{g}_i^j$	the jth constraint in $\mathbf{A}_i$'s design
i	agent index
$j, \bar{j}, k$	variable and noise factor indices
L	cost function
L_c	isolated cost, as a function of a conceptual noise factor
L_d	isolated cost, as a function of a design variable
$\hat{L}_d$	integrated cost, as a function of a design variable
L_{dc}	cost, as a function of a design variable and a conceptual noise factor
$\Delta L_{d_i^j c_i^k}$	maximum cost improvement to be gained by waiting until c_i^k is fixed before fixing d_i^j
$L_{delay,i}$	cost as a function of time delay for $\mathbf{A}_i$
L_{DM}	cost of eliminating a value of a design variable immediately
$L_{TIME,i}$	cost of waiting one time period for $\mathbf{A}_i$
L_{TOTAL}	cost of delaying the whole design project
L_{WAIT}	expected cost, as a function of a design variable, achievable by waiting until all conceptual noises have been fixed before fixing the variable
$\hat{L}_{WAIT}$	true cost, as a function of a design variable, achievable by waiting until all conceptual noises have been fixed before fixing the variable
$L_{\Delta,i}$	total marginal cost to other agents of $\mathbf{A}_i$'s decisions on an interface variable
$l_{\Delta,i}$	marginal cost to $\mathbf{A}_i$ of another agent's decisions on an interface variable
m	number of agents
n	dimension of $\mathbf{D}$
n_i	dimension of $\mathbf{D}_i$
n_i^C	dimension of $\mathbf{C}_i$
$\mathbf{P}_i$	the space containing physical noise factors of $\mathbf{A}_i$
$\boldsymbol{p}_i$	a vector in $\mathbf{P}_i$
t	time
t_0	time period between meetings
t_i	due time for $\mathbf{A}_i$
$\mathbf{X}_i$	the space containing all the factors that affect $\mathbf{A}_i$'s design
$\boldsymbol{x}_i$	a position vector in $\mathbf{X}_i$
$\mathbf{Y}_i$	the space containing the interdependent design variables controlled by $\mathbf{A}_i$
y_i^j	the jth interdependent design variable or one-dimensional subspace of $\mathbf{Y}_i$
ρ	probability density function
$\dim(\cdot)$	dimension of a space
$\mathrm{E}[\cdot]$	expectation
$Pr(\cdot)$	probability
$\backslash$	set of space subtraction
$\mid$	condition

□

References

Ahn, J-S. and Crawford, R. H. 1994. "Complexity Analysis of Computational Engineering Design Processes", *Design Theory and Methodology*, ASME, DE-vol. 68, pp. 205–20.

Aho, A. V., Hopcroft, J. E. and Ullman, J. D. 1983. *Data Structures and Algorithms*, Addison-Wesley, Reading, Mass.

Bazaraa, M. S., Sherali, H. D. and Shetty, C. M. 1993. *Nonlinear Programming*, Wiley, New York.

Birmingham, W. P., Darr, T. P., Durfee, E. H., Ward, A. C. and Wellman, M. P. 1993. "Supporting Mechatronic Design via a Distributed Network of Intelligent Agents", *Workshop on AI in Collaborative Design*. AAAI, July 11–15, 1993, Washington, DC.

Bond, A. H. and Ricci, R. J. 1992. "Cooperation in Aircraft Design", *Research in Engineering Design*, vol. 4, pp. 115–30.

Bradley, S. R. and Agogino, A. M. 1991. "An Intelligent Real Time Design Methodology for Catalog Selection", *Design Theory and Methodology*, vol. 31, pp. 201–8.

Chang, T.-S., Ward, A. C., Lee, J. and Jacox, E. H. 1994. "Conceptual Robustness in Simultaneous Engineering: An Extension of Taguchi's Parameter Design", *Research in Engineering Design*, vol. 6, pp. 211–22.

Clark, K. B. and Fujimoto, T. 1991. *Product Development Performance*, Harvard Business School Press, Boston, Mass.

Consoli, R. D. and Sobieszczanski-Sobieski, J. 1992. "Application of Advanced Multidisciplinary Analysis and Optimization Methods to Vehicle Design Synthesis", *Journal of Aircraft* (Sept.–Oct.), vol. 29, no. 5, pp. 811–18.

Cook, H. E. 1992. "New Avenues to Total Quality Management", *Manufacturing Review* (Dec.), vol. 5, pp. 284–92.

Cutkosky, M. R., Conru, A. B. and Lee, S-H. 1994. "An Agent-based Approach to Concurrent Cable Harness Design", *AIEDAM*, vol. 8, pp. 45–61.

Cutkosky, M. R. and Tenenbaum, J. M. 1990. "Toward a Computational Framework for Concurrent Engineering", *IECON*, IEEE, vol. 1, pp. 700–6.

Darr, T. P. and Birmingham, W. P. 1994. "Automated Design for Concurrent Engineering", *IEEE Expert*, October 1994, vol. 9, No. 5, pp. 35–42.

Eppinger, S. D., Whitney, D. E., Smith, R. P. and Gebala, D. A. 1990. "Organizing the Tasks in Complex Design Projects", *Design Theory and Methology*, DE-vol. 27, pp. 39–46.

Eppinger, S. D., Whitney, D. E. and Gebala, D. A. 1992. "Organizing the Tasks in Complex Design Projects: Development of Tools to Present Design Procedures", *Proceedings, NSF Design and Manufacturing System Conference*, Atlanta, Ga., January 1992.

Finger, S. and Dixon, J. R. 1989. "A Review of Research in Mechanical Engineering Design. Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Processes", *Research in Engineering Design*, vol. 1, pp. 51–67.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass.

Krishnan, V., Eppinger, S. D. and Whitney, D. E. 1991. "Towards a Cooperative Design Methodology: Analysis of Sequential Descision Strategies", *Design Theory and Methodology*, DE-vol. 31, pp. 165–72.

Michelena, N. F. and Agogino, A. M. 1991. "Formal Solution of N-type Taguchi Parameter Design Problems with Stochastic Noise Factors", *Design Theory and Methodology*, DE-vol. 31, pp. 13–20.

Otto, K. N. and Antonsson, E. K. 1993. "Extension to the Taguchi's Method of Product Design", *Journal of Mechanical Design* (March), vol. 115, pp. 5–13.

Papalambros, P. Y. and Wilde, D. J. 1988. *Principles of Optimal Design*, Cambridge University Press, New York.

Siddall, J. N. 1983. *Probabilistic Engineering Design*, Dekker, New York.

Smith, P. G. and Reinertsen, D. G. 1991. *Developing Product in Half the Time*, Van Nostrand Reinhold, New York.

Sobieszczanski-Sobieski, J., James, B. B. and Dovi, A. R. 1985. "Structural Optimization by Multilevel Decomposition", *AIAA Journal* (Nov.), vol. 23, no. 11, pp. 1775–82.

Sobieszczanski-Sobieski, J. and Tulinius, J. 1991. "MDO Can Help Resolve the Designer's Dilemma", *Aerospace America*, September, pp. 32–5, 63.

Taguchi, G. 1986. *Introduction to Quality Engineering*, Krauss International Publications, White Plains, NY.

Taguchi, G. 1987. *System of Experimental Design*, vols. 1 and 2, Krauss International Publications, White Plains, NY.

Taguchi, G. and Clausing, D. 1990. "Robust Quality", *Harvard Business Review* (Jan.–Feb.), vol. 68, pp. 65–75.

Ulrich, K., Sartorius, D., Pearson, S. and Jakiela, M. 1993. "Including the Value of Time in Design-for-Manufacturing Decision Making", *Management Science* (April), vol. 39, no. 4, pp. 429–47.

Vasseur, H., Kurfess, T. R. and Cagan, J. 1993. "Economic Analysis of Quality Innovation in Design and Manufacturing", *Advances in Design Automation*, ASME, DE-vol. 65–2, pp. 495–500.

Von Neumann, J. and Morgenstern, O. 1964. *Theory of Games and Economic Behavior*, Wiley, New York.

Wagner, T. C. 1993. "A General Decomposition Methodology for Optimal System Design", PhD dissertation, University of Michigan.

Wagner, T. C. and Papalambros, P. Y. 1993. "A General Framework for Decomposition Analysis in Optimal Design", *Advances in Design Automation*, ASME, DE-vol. 65–2, pp. 315–25.

Ward, A. C. and Seering, W. 1993a. "The Performance of a Mechanical Design Compiler", *Journal of Mechanical Design* (Sept), vol. 115, pp. 341–5.

Ward, A. C. and Seering, W. 1993b. "Quantitative Inference in a Mechanical Design Compiler", *Journal of Mechanical Design* (March), vol. 115, pp. 29–35.

Ward, A. C., Liker, J., Sobek, D. and Christiano, J. 1995. "The Second Toyota Paradox" accepted by *the Sloan Management Review.*

Wellman, M. P. 1993. "A Market-oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems", *Journal of Artificial Intelligence Research*, vol. 1, pp. 1–23.

Wellman, M. P. 1994. "A Computational Market Model for Distributed Configuration Design", *Proceedings of the 12th National Conference on AI*, AAAI, vol. 1, pp. 401–7. Seattle, WA, 1994.

Wilde, D. J. 1992. "Monotonicity Analysis of Taguchi's Robust Circuit Design Problem", *Journal of Mechanical Design* (Dec), vol. 114, pp. 616–9.

Wilde, D. J. 1991. "A Counterexample to Signal-to-Noise Ratio as a Measure of Design Robustness", *Advances in Design Automation*, DE-vol. 32-1, pp. 233–4.