A study of convergence in decentralized design processes

Received: 28 July 2003 / Accepted: 3 October 2005 / Published online: 25 November 2005
© Springer-Verlag London Limited 2005

Abstract The decomposition and coordination of decisions in the design of complex engineering systems is a great challenge. Companies who design these systems routinely allocate design responsibility of the various subsystems and components to different people, teams or even suppliers. The mechanisms behind this network of decentralized design decisions create difficult management and coordination issues. However, developing efficient design processes is paramount, especially with market pressures and customer expectations. Standard techniques to modeling and solving decentralized design problems typically fail to understand the underlying dynamics of the decentralized processes and therefore result in suboptimal solutions. This paper aims to model and understand the mechanisms and dynamics behind a decentralized set of decisions within a complex design process. By using concepts from the fields of mathematics and economics, including Game Theory and the Cobweb model, we model a simple decentralized design problem and provide efficient solutions. This new approach uses matrix series and linear algebra as tools to determine conditions for convergence of such decentralized design problems. The goal of this paper is to establish the first steps towards understanding the mechanisms of decentralized decision processes. This includes two major steps: studying the convergence characteristics and finding the final equilibrium solution of a decentralized problem. Illustrations of the developments are provided in the form of two decentralized design problems with different underlying behavior.

Keywords Decentralized design · Process convergence · Decomposition · Game Theory · Nash equilibrium

1 Introduction

The focus of this paper is the design of complex engineering systems or those systems that necessitate the decomposition of the system into smaller subsystems in order to reduce the complexity of the design problems. Most of these systems are very large and have a great number of subsystems and components. Companies are under increasing pressure to design them faster due to market pressures and customer expectations. To do so, they have to find new ways to make systems more effective or design processes more efficient. In this paper, we focus on the dynamics of distributed design processes and attempt to understand the fundamental mechanics behind these processes in order to facilitate the decision process between networks of decision makers.

These complex engineering systems are multidisciplinary in nature, and it is therefore impossible for one designer, or even a single design team, to consider the entire system as a single design problem. Typically, in complex systems, breaking it up into smaller units or subsystems will make the system more manageable. This process, known as decomposition, is now seen as a necessary step in design, especially in such complex systems (Krishnamachari and Papalambros 1997; Kusiak and Wang 1993).

The decentralization of decisions is unavoidable in a large organization where having only one centralized decision maker is usually not applicable (Lee and Whang 1999). A more effective way is to delegate decision responsibilities to the appropriate person, team or supplier. Several methods have been developed to deal with the decentralized design systems involving coupled subsystems, with applications in product design (Androulakis and Reklaitis 1999; Wheeler and Narendra 1985), control systems (Tomlin et al. 1998), manufacturing systems (Krothapalli and Deshmukh 1999) and computing architectures (Shahabi and Banaei-Kashani 2002). In fact, decentralization is recommended as a way to speed up product development processes and decrease
the computational time and the complexity of the problem (Prewitt 1998).

While the decomposition of complex problems certainly creates a series of smaller, less complex problems, it also creates several challenging issues associated with the coordination of these less complex problems. The origin of these problems is the fact that the less complex subproblems are usually coupled and dependent upon information from other subproblems. The ideal case would be when a system could be broken up into subsystems without interdependence. Unfortunately, there are usually design variables and parameters that have an influence on several subproblems. Figure 1 illustrates this situation in the case of a simplified aircraft design problem. Only two subproblems are considered here, the aerodynamics and the propulsion subproblems. The aerodynamics design team is responsible for maximizing the lift to drag ratio, \( L/D \), by controlling the wing surface area, \( S \), while the propulsion team wants to minimize the cost of the engine by choosing an appropriate installed thrust, \( T_f \). This system is coupled in the sense that one designer's design variable influences the objective function and constraints of the other design team. The usual approach to solve this type of problem is a sequential approach where designers pass design variables values back and forth until convergence is reached.

The ideal scenario would certainly be when the subproblems were completely uncoupled, as recommended in the Axiomatic Design theory proposed by Suh (1995). Unfortunately, a system with no coupling is typically difficult to achieve. The presence of the coupling generates a number of problems, including the allocation of the system design variables to the subproblems. For instance, if one design variable has an influence on more than one subproblem, there is an issue of which design team or decision maker should have control of this design variable. Previous work has been done on the allocation of the design variables by considering the strengths of the couplings (Bloebaum 1992) or by effectively propagating the desirable top-level design specifications to appropriate subsystems (Michelena et al. 2002). However, this phase of a design process is still critical since it may affect the quality of the final solution (Kim et al. 2000), as well as the convergence of the decentralized design process (Park et al. 2001). Another challenge in coupled systems is the communication barriers that exist between the design teams. The presence of nonlocal variables (design variables controlled by another design team or company) requires a certain level of communication in order to achieve a final optimal design. However, even within the same corporation, perfect information and cooperation is difficult to achieve due to several factors, including the complexity of the design, geographic separation or information privacy.

There are a number of approaches to design coordination in a decentralized environment, including coordination techniques, cooperation techniques, organization models, project management, workflow management and task models (Whitfield et al. 2000). Many of these techniques focus on managing and coordinating a decentralized design process by determining the most effective method of information exchange and the most efficient time to exchange the information. In addition, many of the techniques capitalize on information technology, computing infrastructures and decision support tools to provide effective management and coordination systems (Coates et al. 2000; Whitfield et al. 2002). However, in this work, rather than focus on how and when information is exchanged in distributed design processes, we focus on the fundamental dynamics that dictate the progress and behavior of distributed processes. Unless the distributed design problems are properly conditioned, no coordination or management tool will bring about consensus and convergence in a noncooperative decision process. We believe that by understanding the principles behind the dynamic interactions between designers in a distributed design process, coordination and management tools and systems can be more effectively applied and utilized. Therefore, we focus on decentralized design scenarios where full and efficient exchange of all information among subsystem designers is not possible. In other words, we focus on scenarios where cooperation among designers cannot be elicited by any person or tool. The designers exchange information, but make their own decisions in isolation within the distributed environment.

To overcome these problems associated with noncooperative scenarios, techniques that focus on the interfaces between decision makers, such as Game Theory, can provide new and interesting insights. The foundations of Game Theory lie in the work of Von Neumann and Morgenstern (1944) and Nash (1951). Game Theory provides a mathematical framework that models the interaction between decision makers, also called players. It has been mainly used in the fields of economics and social sciences before finding applications in other areas.

![Fig. 1 Example of coupled subproblems](image-url)

**Subproblem Aero**

\[
\text{Maximize} \\
L/D = f(S,T_f) \\
\text{subject to} \\
g_i(S,T_f)
\]

**Subproblem Prop.**

\[
\text{Maximize} \\
\text{Cost} = f_g(T_p,S) \\
\text{subject to} \\
g_3(T_p,S)
\]
of interest, from the stock exchange to engineering design. Indeed, Game Theory is very useful in distributed decision problems, where designers make decisions that influence one another’s objective function and where the level of communication between those designers is not perfect. Game Theory in engineering design has been and still is an important area of research. The primary goal is to try to improve the quality of the final solution in a multiobjective, distributed design optimization problem (Vincent 1983). Previous work in Game Theory includes work to model the interactions between the designers if several design variables are shared among designers (Lewis and Mistree 1997). In Mistree and Marston (2000), Game Theory is formally presented as a method to help designers make strategic decisions in a scientific way. In Hernandez et al. (2002), distributed collaborative design is viewed as a noncooperative game, and maintenance considerations are introduced into a design problem using concepts from game theory. In Allen (2001), the manufacturability of multi-agent process planning systems is studied using Game Theory concepts. In Rao et al. (1997), noncooperative protocols are studied and the application of Stackelberg leader/follower solutions is shown. In addition, in Chen and Li (2001), a Game Theory approach is used to address and describe a multifunctional team approach for concurrent parametric design. This set of previous works has established a solid foundation for the application of Game Theory in design, but has not directly studied the mechanisms of convergence in a generic decentralized design problem.

This paper uses some concepts from Game Theory, particularly concepts from noncooperative games such as the Nash equilibrium, to study decentralized decision processes in a network of companies or within a large, distributed company. These concepts are used towards the original contribution of this paper, which is a preliminary study of convergence in decentralized problems. In the next section, the general structure of a decentralized design process is presented, and two motivating examples are discussed to establish the premise for the work in this paper. Then, we develop our approach for modeling and solving a generic decentralized problem and apply it to two simple decentralized decision problems.

2 Motivating examples

Consider the example of a system that has been decomposed into two smaller subsystems. We assume that there is some communication between them and that the decision makers responsible for each subsystem (whether they represent an individual, a team or a company) are willing to give only the value of the design variables they are controlling. In other words, each decision maker will share his/her decision, whether it is to choose a certain material or to set the length of a beam to a certain value. A classical approach to this problem is to have one decision maker begin the process by solving his/her design problem and then passing the information to the other decision maker. Then, the second decision maker solves his/her problem and passes the information back to the first decision maker. This process continues until convergence is reached. This iterative process is shown in Fig. 2, where designer 1 attempts to minimize objective function \( F_1 \) and controls design variable \( x \), and designer 2 attempts to minimize objective function \( F_2 \) and controls design variable \( y \).

For any choice of \( y \) by designer 2, the first designer would rationally choose \( x \) that minimizes \( F_1 \). Such a pair is known as a rational reaction set (RRS) for the first designer. The RRS of a player is a function that embodies his/her reactions to decisions made by other players (Rao et al. 1997). An RRS for the second designer can be similarly defined. If these two sets intersect, the point of intersection is known as the Nash solution point, which has a property of being an equilibrium point. A point is said to be a Nash equilibrium or a Nash solution if no designer can improve unilaterally his/her objective function (Thompson 1953). The mathematical formulation of the Nash equilibrium for two objectives and two design variables is as follows. A strategy pair \((x_N, y_N)\) is a Nash solution if

\[
\begin{align*}
F_1(x_N, y_N) &= \min_x F_1(x, y) \\
F_2(x_N, y_N) &= \min_y F_2(x, y)
\end{align*}
\]

This solution has the property of being individually stable, but is not necessarily collectively stable, meaning that, at this point, each designer will perceive the design point to be optimal whereas the solution is not necessarily collectively optimal (Friedman 1986). This is because any unilateral decision to change a design variable value by either designer cannot, by definition, result in a better objective function value for the designer who makes the change.

The importance of the existence of at least one Nash equilibrium point for a given problem is significant. Figure 2 shows the iterative design process followed by the designers. If there is no Nash equilibrium, the designers will always think that they can improve their objective function. Therefore, there will never be convergence, and the iterative approach would theoretically have no end. It is only when the designers approach a Nash equilibrium point that the changes in the values of the design variables will become smaller, and convergence will occur, bringing an end to the iterative process.

This convergence phenomenon, as well as the importance of the RRSs, is best illustrated by means of two examples, adopted from Vincent (1983). Figure 3 illustrates a situation in which the designers have the following objective functions:

\[
F_1 = x^2 - 3x + xy \\
F_2 = \frac{x^2}{2} - xy
\]

(2)
where designer 1 controls $x$ and minimizes $F_1$, and designer 2 controls $y$ and minimizes $F_2$, with $x \geq 0$ and $y \geq 0$. Figure 3 illustrates the RRSs for each designer as well as the first iterations of the process depicted in Fig. 2.

According to Fig. 3, the designers move back and forth between their RRSs. There are several ways to find the RRSs, and some research has been done to find efficient techniques to determine RRSs for large and complex problems (Lewis and Mistree 2001). However, in our case of unconstrained unimodal optimization problems (Eq. 2), a simple way to find the RRS of a designer is to set the partial derivative of their objective function with respect to their design variable to zero, as shown in Eq. 3. Solving this equation will indeed give the global minimum of one designer's objective function in terms of the other designer's design variable:

$$\frac{\partial F_1}{\partial x} = 2x - 3 + y = 0 \quad \frac{\partial F_2}{\partial y} = y - x = 0$$

A simple equation for the RRS of each designer in terms of the other designer’s variable is found:

$$\text{Fig. 2 Algorithm of a sequential approach}$$

$$\text{Fig. 3 A convergent decentralized design example}$$
Suppose that designer 1 makes the first tentative decision to choose the global minimum of \( F_1 \) (1.5,0). Designer 2, knowing \( x=1.5 \), would choose \( y=1.5 \) in order to move the solution to his/her RRS. This design is then passed back to designer 1 who, using \( y=1.5 \), would choose \( x=0.75 \) in order to move the solution to his/her RRS. By continuing this iterative process described in Fig. 2, the solution converges to the Nash solution:

\[
(x_{N}, y_{N}) = (1, 1) \quad (F_{1N}, F_{2N}) = (-1, -0.5).
\]

However, this Nash solution is nonoptimal since there are some points in the design space where both designers could improve their objectives. For example, at \( x=2 \) and \( y=0.33 \), the values of the two objective functions are:

\[
F_1 = -1.33, \quad F_2 = -0.61,
\]

which are better for both designers. Therefore, the Nash solution is dominated by this point since both objectives have been improved. The points of the design space which are not dominated are known as Pareto-optimal solutions, a concept well known in multiobjective optimization (Pareto 1906).

Therefore, when using this iterative decision process in a decentralized environment, the final solution is not necessarily optimal since both objectives could be improved. But what is perhaps even more surprising is that this process does not necessarily have to converge to the Nash solution. Consider the case where two designers have the following objective functions:

\[
F_1 = \frac{x^2}{4} - 1.5x + xy \\
F_2 = \frac{y^2}{2} - xy
\]

With these objective functions, whichever designer goes first and whatever the starting point is, the decentralized decision system will always diverge. As an example, designer 2 starts with the tentative solution \((x=0, y=0.8)\) and passes this information to designer 1 who adjusts \( x \). This information is then passed back to designer 2 to adjust \( y \), and so on as described by the algorithm in Fig. 2. Carrying this process out results in a divergent process as illustrated in Fig. 4.

This issue of a divergent decentralized decision process is challenging. Indeed, in the case of convergence, designers agree on a final combination of the design variables even though the solution is not necessarily optimal. However, in the case of divergence, designers will never agree on a final design since one of the designers will always be able to change the value of their design variables and improve their objective function.

How the two designers might go about choosing the final design is then difficult to predict. But in the absence of any additional information or intervention by a third party, it seems obvious that the solution will not be optimal.

Since this problem of convergence is crucial to a design process, a way to determine whether there is convergence or not would be beneficial to studying the dynamics of decentralized decision processes. The quality of the solution is also important, but the process has to first converge before solution quality even becomes relevant. Thus, in this paper, we focus on the dynamics of the convergence of the decentralized system. Some of the preliminary work in this area is presented in Chanron and Lewis (2003), including an investigation of simple decentralized systems with two designers who each control only one variable (e.g., the systems represented in Eqs. 2, 5). This paper goes further by developing convergence conditions for problems where the designers each control an arbitrary number of design variables. This represents a necessary and critical step towards developing an approach to determine the convergence behavior of any decentralized decision problem. The next section presents the basic mathematical formulation and develops the convergence criteria.

3 Development of convergence criteria

To study the basic mechanism behind decentralized processes, we focus on a decentralized decision problem with only two designers, each of them controlling an indefinite number of design variables. Designer 1 has control of the vector of design variables \( x \) while designer 2 has control of the vector of design variables \( y \). A general formulation for the objective functions is used, and the goal of this section is to find some conditions under which there will be convergence and, in the case of convergence, to find the final values of the design variables.

Two assumptions are made in order to simplify the calculations: the objective functions are quadratic and the optimization problem is unconstrained. These assumptions look rather restrictive compared to the complexity of typical design problems. However, any optimization problem can be converted to an unconstrained problem by the use of a penalty function (Vanderplaats 1999) and many algorithms use this approach, including many genetic algorithms, simulated annealing and other heuristic methods. Moreover, many optimization techniques and algorithms use simplifications or approximations. Some concepts have been developed that allow the use of multiple approximation models of different types simultaneously in one optimization problem (Golovidov et al. 1998). In addition, quadratic approximations are also used in optimization often since they are not as limiting as linear approximations but are not too complex either (Chen et al.)
Fig. 4 A divergent decentralized design example

1996; Pérez et al. 2002; Renaud and Gabriele 1994; Wang et al. 2001). Therefore, the assumptions are justified in order to understand the dynamics of a general decentralized process case and as a starting point in the investigations. However, to be applicable to realistic decentralized design problems, constraints and highly nonlinear objective functions need to be included in the approach. This is the logical next step in this work once the basic mechanisms of decentralized convergence are understood.

The general quadratic form for the two objective functions is shown in Eq. 6 with designer 1 controlling the vector \( \mathbf{x} \) (size \( n \times 1 \)) and designer 2 controlling the vector \( \mathbf{y} \) (size \( p \times 1 \)). \( \mathbf{A} \) and \( \mathbf{Q} \) are square matrices of size \( m \times m \) and \( p \times p \); the reader is referred to Chanron (2002) for more information about matrix sizes:

\[
\begin{align*}
F_1 &= \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{y}^T \mathbf{B} \mathbf{y} + \mathbf{y}^T \mathbf{C} \mathbf{x} + \mathbf{D} \mathbf{x} + \mathbf{E} \mathbf{y} + \mathbf{F} \\
F_2 &= \mathbf{y}^T \mathbf{Q} \mathbf{y} + \mathbf{x}^T \mathbf{R} \mathbf{x} + \mathbf{x}^T \mathbf{S} \mathbf{y} + \mathbf{T} \mathbf{y} + \mathbf{U} \mathbf{x} + \mathbf{V} \\
\end{align*}
\]

(6)

The first task is to find the RRSs of both designers. Indeed, as seen previously, the designers will always return to their RRSs in order to minimize their objective functions. For example, the RRS of designer 1 gives the value of \( \mathbf{x} \) that minimizes \( F_1 \) as a function of \( \mathbf{y} \).

If the matrices \( \mathbf{A} \) and \( \mathbf{Q} \) are nonsingular, a unique expression can be found for the RRSs by finding the vector of partial derivatives with respect to each variable vector and setting it equal to zero. These expressions are presented in Eq. 7. We discuss the implications of the singularity of the matrices \( \mathbf{A} \) and \( \mathbf{Q} \) in the next section:

\[
\begin{align*}
\text{RRS1: } \mathbf{x} &= -\frac{1}{2} \mathbf{A}^{-1} \mathbf{C}^T \mathbf{y} - \frac{1}{2} \mathbf{A}^{-1} \mathbf{D}^T \\
\text{RRS2: } \mathbf{y} &= -\frac{1}{2} \mathbf{Q}^{-1} \mathbf{S}^T \mathbf{x} - \frac{1}{2} \mathbf{Q}^{-1} \mathbf{T}^T \\
\end{align*}
\]

(7)

These are the generic forms of the RRSs. In order to study the behavior of these RRSs, we make an analogy between their behavior and the behavior of supply and demand curves in microeconomic theory. As shown in Fig. 5, the behavior of typical supply and demand curves is similar to the behavior of the RRSs with respect to monotonicity. A concept from microeconomic theory called the Cobweb model (Pindyck and Rubinfeld 2001) describes how economic systems with given supply and demand curves converge to stable supply and demand levels. This stable equilibrium in supply and demand is the same as the Nash equilibrium in Fig. 3 at the intersection of the RRSs (Branch 2000).

The supply and demand convergence in economic systems could take years and typically is modeled as a time series where the next year’s shift in supply and/or demand is based on the supply and demand in the current year. We therefore model the RRSs as a set of coupled series, \( (x_n) \) and \( (y_n) \), where one series depends on the decision made by the other designer in the previous time step. This has already been successfully used in Chanron and Lewis (2003) in the case of single design variables. This series notation is shown in Eq. 8:

\[
\begin{align*}
x_{n+1} &= -\frac{1}{2} \mathbf{A}^{-1} \mathbf{C}^T y_n - \frac{1}{2} \mathbf{A}^{-1} \mathbf{D}^T \\
y_{n+1} &= -\frac{1}{2} \mathbf{Q}^{-1} \mathbf{S}^T x_n - \frac{1}{2} \mathbf{Q}^{-1} \mathbf{T}^T \\
\end{align*}
\]

(8)
The convergence (or divergence) of these series \((x_n)\) and \((y_n)\) corresponds to the convergence of the iterative process. Evaluating their convergence is the goal of this section. To do so, it is first important to notice that they have the same behavior. In other words, if \((x_n)\) converges, \((y_n)\) will converge, and if \((x_n)\) diverges, \((y_n)\) will diverge. Indeed, if one of them is converging to a finite value and the other one increases to infinity as \(n\) tends to infinity, then Eq. 8 will not be valid. Thus, studying only the behavior of \((x_n)\) will be adequate to find the convergence of the iterative process. However, we here deal with vectors of design variables, instead of scalars as in previous work (Chanron and Lewis 2003). Therefore, the series \((x_n)\) is a matrix series. Studying its convergence is more complex than studying the convergence of a scalar series, and it requires notions of linear algebra.

First of all, in order to simplify the notations, one of the recurrent expressions needs to be fully defined. In many scientific fields, the representation and properties of a system are dependent on properties of simpler characteristic elements: state-space for a linear system and characteristic equation for a dynamic system or a differential equation. We therefore define \(K\), as shown in Eq. 9, to be the characteristic matrix of a decentralized problem with two designers. The properties of this matrix capture and to a large extent dictate the properties of the entire decentralized process:

\[
K = \frac{1}{2}A^{-1}C^TQ^{-1}S^T.
\]  

(9)

This matrix has interesting properties that influences the convergence of the decentralized decision process, but its most important feature is that it is a square matrix of size \(n \times n\) (Chanron 2002). Using this matrix \(K\), a general formulation for the series \((x_n)\) is found and shown in Eq. 10. This formulation has been found by recursively applying Eq. 8, and using the recurrent appearance of \(K\) in the expression of \((x_n)\):

\[
x_{n+1} = K^2x_1 + \sum_{k=0}^{n-1} K^k \left[ \frac{1}{4}A^{-1}C^TQ^{-1}T^T - \frac{1}{2}A^{-1}D^T \right].
\]

(10)

Using mathematical induction, this formulation is proven. Starting with the case of \(n=0\), we immediately have \(x_1 = x_1\). We next consider the case when \(n=1\). However, since we are investigating an iterative series that inherently consists of two steps, one for each designer represented in Eq. 8, the case \(n=1\) is not meaningful. In order to have one complete cycle, we take \(n=2\). For \(n=2\) and using Eq. 8:

\[
x_3 = -\frac{1}{2}A^{-1}C^TQ^{-1}S^T x_1 + \frac{1}{4}A^{-1}C^TQ^{-1}T^T - \frac{1}{2}A^{-1}D^T
\]

\[
= Kx_1 + \left[ \frac{1}{4}A^{-1}C^TQ^{-1}T^T - \frac{1}{2}A^{-1}D^T \right],
\]

which is the same form as given in Eq. 10 with \(n=2\). We then assume Eq. 10 to be true for \((n-1)\) and prove Eq. 10 to be true at the next logical step \((n+1)\). We then have the expression for \((x_{n-1})\):

\[
x_{n-1} = K^{n-1}x_1 + \sum_{k=0}^{n-2} K^k \left[ \frac{1}{4}A^{-1}C^TQ^{-1}T^T - \frac{1}{2}A^{-1}D^T \right].
\]

(12)
Computing the expression of \( (x_{n+1}) \) by successively using Eqs. 8 and 12 gives

\[
x_{n+1} = \frac{1}{2} A^{-1} C^T y_n + \frac{1}{2} A^{-1} D^T
= K x_n + K \sum_{k=0}^{n} K^k \left[ \frac{1}{4} A^{-1} C^T Q^{-1} T^T + \frac{1}{2} A^{-1} D^T \right]
+ \left[ \frac{1}{4} A^{-1} C^T Q^{-1} T^T + \frac{1}{2} A^{-1} D^T \right]
= K^2 x_1 + \sum_{k=0}^{n-1} K^k \left[ \frac{1}{4} A^{-1} C^T Q^{-1} T^T - \frac{1}{2} A^{-1} D^T \right], \quad (13)
\]

which is exactly Eq. 10, thus concluding the mathematical induction proof of Eq. 10.

From this general expression of the series \((x_n)\), we can determine the condition for convergence of this series. This type of matrix geometric series is called a Neumann series and defined by the following equation (Meyer 2001):

\[
I + A + A^2 + \cdots + A^n = \sum_{k=0}^{n} A^k, \quad (14)
\]

To study the convergence of the Neumann series, we use one of the consequences of Schur's Theorem presented in Strang (1988):

For any square matrix \(A\), in order that \(\|A\|^k \to 0\) \(|k| \to \infty\) in (every) matrix norm, it is necessary and sufficient that \(r_d(A) < 1\).

where \(r_d(A)\) denotes the spectral radius of \(A\) and represents the maximum absolute value of the eigenvalues of \(A\) (Walsh 2000):

\[
r_d(A) = \max \{ |\lambda| : \lambda \in \sigma(A) \}. \quad (15)
\]

Using these definitions, the convergence for any Neumann series can be determined using the following condition (Fei 2002):

The Neumann series converges if and only if \(r_d(A) < 1\).

In the case of convergence, the limit of the series is given by Eq. 16:

\[
\sum_{k=0}^{\infty} A^k = (I - A)^{-1}. \quad (16)
\]

We can now apply these definitions to study the convergence of Eq. 10. The series \((x_n)\) is the sum of two different terms, which both need to converge for the overall series to converge. They both have the same convergence criterion, as they are both a Neumann series with the same characteristic matrix \(K\). Therefore, a simple convergence criterion for this series as well as for the decentralized decision problem is shown in Eq. 17:

\[
(x_n) \text{ converges if and only if } r_d(K) < 1. \quad (17)
\]

This convergence criterion for the decentralized decision process represents the first primary contribution of this work. The second contribution consists of finding the equilibrium point of this process. This is done by taking the limits of the series introduced in Eq. 8 and using the property introduced in Eq. 16. We denote \(x^*\) and \(y^*\), the limits of the series \((x_n)\) and \((y_n)\), respectively, and their expression is given in Eq. 18:

\[
\begin{align*}
x^* &= (I - K)^{-1} \left[ \frac{1}{2} A^{-1} C^T Q^{-1} T^T - \frac{1}{2} A^{-1} D^T \right] \\
y^* &= \frac{1}{2} Q^{-1} S^T (I - K)^{-1} A^{-1} D^T \\
&\quad - \frac{1}{2} \left[ \frac{1}{4} Q^{-1} S^T (I - K)^{-1} A^{-1} D^T + 1 \right] Q^{-1} T^T
\end{align*}
\]

The convergence criterion as well as the expressions for the final values of the design variables seem rather complicated, but are only functions of coefficient matrices from the problem formulation. A better understanding of those equations can also be gained by examining the results when designers control only one design variable (Chanran and Lewis 2003), which is only a special case of the results presented here. Matrices are replaced by scalars, and inverses by ratios, making the equations much simpler. The problem formulation simplifies to Eq. 19:

\[
F_1 = ax^2 + by^2 + cxy + dx + ey + f
F_2 = gy^2 + bx^2 + cyx + dy + ex + f
\]

Similar to the characteristic matrix \(K\) defined in Eq. 9, a convergence ratio, \(r\), and subsequent criterion is defined for this decentralized formulation in Eq. 20:

\[
r = \left| \frac{G'}{4ax} \right| < 1. \quad (20)
\]

This convergence criterion is a simple case of the more general condition of Eq. 17. In addition, the limit of the design variables as \(n \to \infty\) is given in Eq. 21:

\[
\begin{align*}
x^* &= \frac{e_1 - d_2}{4ax - c_1} \\
y^* &= \frac{e_2 - a_2}{4ax - c_1}
\end{align*}
\]

By examining both the generalized and simplified conditions, a thorough understanding of the behavior of any decentralized system under our operating assumptions is developed. Further implications of these criteria are presented in the next section, including a discussion of the assumptions and unique cases of the convergence criterion.

4 Further implications of the criteria

The conditions presented in the previous section have given the behavior in the general case, but some particular cases require more investigation, because they either create certain problems or have some interesting
properties. One of the conditions necessary to find a unique equation for the RRSs of the two designers is that matrices $A$ and $Q$ are nonsingular. In this section, we look at the case when $A$ is singular, knowing that the same analysis can be made for the case when $Q$ is singular. There are two primary cases when $A$ can be singular. These cases are studied next and recommendations are made on how to alter the problem formulation to make $A$ singular without affecting the underlying dynamics in the decentralized decision process.

Case 1: When there are no cross terms between the design variables of designer 1 in $F_1$ (e.g., $x_1x_2$, $x_1x_3$, $x_1x_4$), $A$ is a diagonal matrix. $A$ would therefore be singular only if one of the diagonal coefficients is equal to zero. This would occur when no squared term exists for one of the $x_i$ variables. In this case, with no squared term or cross terms, the terms in $F_1$ that include $x_i$ are linear (e.g., $2x_1$) or cross terms with one of the $y_i$ variables in $y$ (e.g., $x_1y_2$). When designer 1 makes his/her decision by setting the gradient of $F_1$ to zero, $x_i$ will not be present in the equation. This is because the partial derivatives of the linear terms in $x_i$ are constant and the partial derivatives of the cross terms are only functions of $y_i$. Therefore the design variable $x_i$ does not influence the dynamics of the decentralized system, as the choice of $x_i$ is independent of the design variables $y_i$. Since the goal of this paper is to study the dynamic relationships between the designers, $x_i$ can be removed from the problem formulation (e.g., set temporarily constant) without influencing the dynamics between the designers (i.e., the convergence or divergence of the design process). This will result in a nonsingular matrix $A$, allowing the method developed in this paper to be applied.

Case 2: When cross terms exist between the design variables in $x$, $A$ is no longer a diagonal matrix and for $A$ to be singular, one of the rows of $A$ must be a linear combination of the others. In that case, a change of variable can be made in order to decrease the number of design variables. To confirm this consider an example where designer 1 has control of three design variables and the matrix $A$ is the following:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 3 \end{pmatrix}.$$  \hspace{1cm} (22)

In this matrix the second row is simply the first row multiplied by two. Therefore, the determinant of this matrix is 0. However, an investigation of the representation of $F_1$ in Eq. 23 allows for a change of variable:

$$F_1 = x_1^2 + 4x_2^2 + 3x_3^2 + 4x_1x_2 + 6x_1x_3 + 12x_2x_3$$

$$= (x_1 + 2x_2)^2 + 3x_3^2 + 6(x_1 + 2x_2)x_3.$$ \hspace{1cm} (23)

A simple change of variable can be made: $x' = (x_1 + 2x_2)$ decreasing the number of variables to 2 and giving the following 2x2 matrix for $A$:

$$A = \begin{pmatrix} 1 & 3 \\ 3 & 3 \end{pmatrix}.$$ \hspace{1cm} (24)

$A$ is now a nonsingular symmetric matrix, allowing Eqs. 17 and 18 to be directly applied.

Another unique case of the characteristic matrix $K$ in Eq. 9 occurs when the matrix $C$ or $S$ is equal to 0. This would mean that either objective function 1 or objective function 2 does not have any term that includes both $x$ and $y$. If $C = 0$, then $F_1$ is the sum of terms only dependent on $x$, and terms only dependent on $y$. Therefore, the equation of the RRS of designer 1, which is found by taking the derivative of $F_1$ with respect to $x$, will be a function of only $x$. That means for whatever values of $y$ given by designer 2, designer 1 will always return the same value for his/her design variables $x$. Therefore, the classic iterative process will converge in less than two iterations. This result is validated by checking the convergence criterion in Eq. 17. Since $C$ or $S$ is equal to 0, $K$ is equal to 0, and therefore all its eigenvalues (and by definition its spectral radius) will be 0.

In the following section two example problems are investigated to validate the results and explore their application.

5 Example problems

In the first example problem, the two designers are minimizing the objective functions shown in Eqs. 25 and 26. These objective functions could be the initial objective function augmented with a penalty function, in the case of constrained optimization. We consider that $x$ has two components, while $y$ has three components, which is the number of design variables that the designers are controlling:

$$F_1 = 2x_1^2 + x_2^2 + 2x_1x_2 + x_1y_1 + 2x_1y_2 + x_2y_3$$

$$+ x_2y_3 - 2x_1 - 3x_2$$

$$= x^T\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 0 \end{pmatrix}x + y^T\begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}x + (-2 - 3)x.$$ \hspace{1cm} (25)

$$F_2 = 2y_1^2 + y_2^2 + \frac{1}{3}y_3^2 + y_1x_1 - y_2x_2 - 2y_1x_1 + 3y_1 + 2y_2 + 3y_3$$

$$= y^T\begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}y + x^T\begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}y + (3 2 2)y.$$ \hspace{1cm} (26)

From these equations, it is difficult to predict without carrying out a number of solution iterations if the decentralized decision system will converge or diverge. Therefore, we apply the approach developed in this paper to investigate the convergence behavior of this process. The first step is to simulate the iterations that designers would go through in an iterative, sequential
approach. Table 1 summarizes the results of the first few iterations and shows the values of the design variables at each of the iterations. The values appear to be oscillating around their final values, and if the process were to continue it may converge.

The final values of the design variables, as well as the number of iterations required to converge, are dependent on the final precision required by the designers. Table 2 summarizes convergence results for three different accuracy requirements. These results show that the process would converge to an acceptable final solution between the designers, but the exact final point is a function of the precision required. It is interesting to notice that the number of iterations required is rather large for this relatively simple problem formulation.

Each iteration represents a certain amount of information passed back and forth between the subsystems. Therefore at every iteration, resources such as time and money are being expended. These costly iterations can be avoided by simply checking the convergence criteria developed in Sect. 3. As suggested in Eq. 17, the eigenvalues and the spectral radius of the characteristic matrix $K$ have to be computed to determine whether or not the process will converge:

$$\text{Eigenvalues of } K = \left\{ -0.6875 + 0.496i, -0.6875 - 0.496i \right\}$$

$$r_e(K) = 0.8478.$$  

Since the spectral radius of the characteristic matrix is less than 1, Eq. 17 assures that the decentralized decision process will converge. Therefore, the final values of the design variables can be directly found, using Eq. 18 and are shown in Eq. 27:

$$x^* = \begin{pmatrix} 2.1717 \\ -1.1313 \end{pmatrix} \quad \text{and} \quad y^* = \begin{pmatrix} -1.2929 \\ -1.5657 \\ 3.5152 \end{pmatrix}. \quad (27)$$

The advantage of this technique becomes clear. Not only can the convergence of the centralized decision process be predicted, but also the final equilibrium values of the design variables can be found, avoiding a large number of costly iterations.

What is perhaps even more interesting is that a slight change in the initial objective functions can alter the outcome of the decentralized process. The second case study investigates a divergent process with the following objective functions:

$$F_1 = 2x_1^2 + x_3^2 + 2x_1x_3 + 3x_1y_1 - 2x_1x_3 + 2x_1y_3 + 2x_2 - x_3y_3 + 2x_1 + 4x_2$$

$$= \mathbf{x}^T \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \mathbf{x} + \mathbf{y}^T \begin{pmatrix} 3 & 0 \\ -2 & 1 \end{pmatrix} \mathbf{x} + (2, 4) \mathbf{x}. \quad (28)$$

$$F_2 = 2x_1^2 + y_3^2 + x_3^2 + y_1x_1 - 3x_1x_3 + 4y_1x_1 - y_3x_2$$

$$= \mathbf{y}^T \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \mathbf{y} + \mathbf{x}^T \begin{pmatrix} 1 & 4 & -2 \\ -3 & -1 & 0 \end{pmatrix} \mathbf{y} + (1, 2, 3) \mathbf{y}. \quad (29)$$

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Values of $x$</th>
<th>Values of $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.5</td>
<td>2.4375</td>
</tr>
</tbody>
</table>
| 2 | 0  
| 3 | 0  
| 4 | 0  
| 5 | 0  
| 6 | 0  
| 7 | 0  
| 8 | 0  
| 9 | 0  
| 10 | 0  |

Table 2 Number of iterations depending on the accuracy desired

<table>
<thead>
<tr>
<th>Precision</th>
<th>Number of iterations</th>
<th>Values of $x$</th>
<th>Values of $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>0.1</td>
<td>36</td>
<td>2.1231</td>
<td>-1.0772</td>
</tr>
<tr>
<td>0.01</td>
<td>66</td>
<td>2.1678</td>
<td>-1.1276</td>
</tr>
<tr>
<td>0.001</td>
<td>96</td>
<td>2.1714</td>
<td>-1.1311</td>
</tr>
</tbody>
</table>
Table 3 shows the values of the design variables for the first few iterations between the designers.

Table 3 clearly shows that the values of the design variables become larger and larger, and that the process will eventually diverge without finding an equilibrium. In that situation, the designers would fail to agree on a final design since one will always have the impression that his/her objective function can be improved. This could have been easily anticipated by using the series convergence criteria. The eigenvalues and spectral radius of the characteristic matrix are calculated as follows:

\[ \text{Eigenvalues of } \mathbf{K} = \{-6.5873, -0.4127\} \]

\[ r_x(\mathbf{K}) = 6.5873. \]

The spectral radius is greater than 1, thus the process will diverge. This is coherent with the expected result, and therefore validates the convergence criterion presented in this paper.

6 Application of the method

The results presented in this paper represent the next logical step for determining convergence of distributed design processes, building upon the initial fundamental study presented in Chanron and Lewis (2003). While the proposed method is still theoretical in nature, there are a number of applications in engineering design of the fundamental results of this work:

- A primary assumption is that mathematical representations of the objective functions of each design or design team exist. If the form of the objective functions is not explicitly known, then in order to apply the method, metamodels of the functions would need to be constructed. Then, a convergence study can be conducted on the approximated system, giving the designers some valuable insight into the predicted behavior of the decentralized system.

- Direct application of the method can be used to study the convergence of more complex problems, such as decentralized design problems involving a large number of designers (Chanron and Lewis 2004), highly nonlinear objective functions (Chanron et al. 2005) and multiple objectives for a single designer. These three factors increase dramatically the complexity of the problem formulation, and therefore the mathematical results presented here are essential for building the convergence criteria for these more complex, yet much more realistic problems.

- The proposed method can also be applied directly to existing design processes. In the conceptual design phase of the design process, the decision makers are not necessarily distributed geographically, but disciplinary design teams are often asked to interact, while having a mathematical representation of their objective function. As shown in the divergent case study presented in the previous section, the convergence criteria could avoid long, costly and essentially useless iterations between designers. This time and effort could rather be used to find another decomposition of the design process that would result in a convergent process. In a convergent system, the time and effort could be used to develop the converged solution in more detail or for other more detailed design tasks.

- Although the results presented in this paper are theoretical, they are necessary to understand and describe the dynamics of decentralized decision design problems. As Tuftes puts it: “An essential analytic task in making decisions based on evidence is to understand how things work—mechanism, trade-offs, process and dynamics, cause and effect” (Tuftes 1997). Therefore, the theoretical knowledge developed in this paper can be used to design new decision-making support tools specifically for decentralized design. For instance, based on the fundamentals of this work, preliminary research has been done on a new design process for distributed design and the results presented here offer new opportunities for future development.

7 Conclusions

Most engineering systems are multidisciplinary in nature and therefore require knowledge from several design

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$x_1$</th>
<th>$x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>-3</td>
</tr>
<tr>
<td>1</td>
<td>0.625</td>
<td>-1.125</td>
</tr>
<tr>
<td>2</td>
<td>2.5938</td>
<td>-4.5</td>
</tr>
<tr>
<td>3</td>
<td>-10.168</td>
<td>14.0273</td>
</tr>
<tr>
<td>4</td>
<td>73.8115</td>
<td>-106.4883</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Values of $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>-2.75</td>
</tr>
<tr>
<td>-1.25</td>
</tr>
<tr>
<td>-4.2734</td>
</tr>
<tr>
<td>12.8125</td>
</tr>
<tr>
<td>-98.5691</td>
</tr>
</tbody>
</table>
teams. This, along with other constraints, usually forces the decentralization of decisions. While centralization of decisions is a preferred approach from a technical perspective, in this paper, we study the scenario where centralization is not possible because of geography, cost, time, resources, organizational structure, etc. We believe that this happens in many product design processes as complete centralization and communication among design groups, engineering teams, suppliers, manufacturers, and other relevant parties are not feasible. Therefore, the decision makers involved in a design process need to understand the fundamental mechanisms of this process in order to find a final optimal design. A new and effective approach to this process is described in this paper, based upon a formal mathematical formulation, and concepts from Game Theory, linear algebra, matrix series theory, and the Cobweb model. The convergence of such systems as well as the final values of the design variables are easily found and understood using this method. This method has been validated on two case studies and has shown to dramatically increase the efficiency of the coordination and convergence process by avoiding a large number of costly iterations between the decision makers. Even though in industrial design problems the centralized designers may not have complete knowledge of the other designers’ design objectives, we believe that studying the underlying dynamics of the process will help either upper level managers or the decision makers themselves make better decisions when decomposing, modeling and solving complex design problems. In addition, by understanding the fundamental dynamics, coordination decision support tools and infrastructures can be more effectively applied.

The future work of this research will concentrate on increasing the complexity of the decentralized model. This will involve several steps, including handling problems with more than two designers, dealing with highly nonlinear objective functions, and incorporating constraints in the problem formulation. In addition, work will focus on investigating the relationship between Nash equilibrium solutions and nondominated Pareto solutions.

Acknowledgements We would like to thank the National Science Foundation, grant DMII-9875706, for their support of this research.

References

Allen B (2001) Game theoretic models of search in multi-agent process planning systems. In: Special interest group on manufacturing (SIGMAN) workshop, international joint conferences on artificial intelligence


Renaud JE, Gabriele GA (1994) Approximation in non-hierarchi-
cal system optimization. AIAA J 32(1):198–205


Vanderplaats GN (1999) Numerical optimization techniques for engineering design, 3rd edn. VK & D


