Predicting the Performance of a Design Team Using a Markov Chain Model

Jeffrey W. Herrmann

Abstract—**When faced with a complex design problem, a design team may separate it into subproblems. We would like to know when this approach is superior and how subproblems should be assigned to team members. We created mathematical models of searches that represent bounded rational decision-makers ("agents") solving a design problem. These discrete-time Markov chains were used to calculate the probability distribution of the value of the solution found and the expected number of steps required. We evaluated the performance of two- and three-agent teams who used two approaches to solve design problems. In the "all-at-once" approach, they search the entire set of solutions. In the "separation" approach, they separate the problem into two subproblems. Three stopping rules and two different types of collaboration were modeled. Using a separation increases the likelihood of finding a high-value solution when high-value solutions are less likely. The optimal assignment of team members to subproblems depended upon the distribution of values in the solution space. These results suggest that more effort should be spent developing better concepts when high-quality concepts are rare. When concepts have similar performance, more effort should be spent searching for better designs that implement the selected concept.**

Index Terms—**New product development, optimization, organizational decision processes, organizational design, Queuing/Markov analysis.**

I. Introduction

DESIGNING a product or system can be viewed as a problem-solving activity in which the design team seeks (searches for) a solution that satisfies the various stakeholders' requirements (which form constraints) and has sufficiently good performance. In practice, teams often separate a complex design problem into subproblems, and then, solve each subproblem instead of tackling the complete problem all-at-once. This approach is a natural strategy, given the limitations that human decision makers have. It is especially convenient in a team setting; different subproblems can be assigned to different members of the team.

If human decision makers were able to optimize, then separating a problem into subproblems would usually lead to solutions that are inferior to those found by solving the problem all-at-once (only in certain conditions will the optimal solutions to the subproblems form an optimal solution to the complete problem).

It is well known, however, that real-world decision makers cannot optimize because of limits on their problem-solving capacity. This concept is known as "bounded rationality" [1]. Bounded rationality reflects the observation that, in most real-world cases, decision makers have limited information and limited computational capabilities for finding and evaluating alternatives and choosing among them, which prevents complete and perfect optimization [2]–[4]. Thus, the separation into subproblems is a reasonable approach to find high-quality solutions.

The study described in this paper was motivated by the following questions.

1) When is separating a better approach than solving the design problem all-at-once?
2) What is the best way to assign subproblems to the members of a team?

This paper presents the results of a study that considered a class of generic design problems, modeled teams of bounded rational decision makers, and evaluated the quality of the solutions that these teams found using different solution approaches. The study was intended to provide insights into the relative performance of the solution approaches. Note that although this study did not consider the process of problem formulation (sensemaking) [5], [6], it did consider the impact of the problem formulation on solution quality.

A novel feature of this study is the use of Markov chains to model the search of a bounded rational decision maker who is solving a design problem. These models can be used to calculate, without resorting to computational simulation, the probability distribution of the value of the solution that the decision maker will find and the expected number of steps required. We created different models by varying the stopping rule and the type of collaboration.

The remainder of the paper proceeds by reviewing the related work on modeling teams of bounded rational decision makers and describing the design problems that we studied. This paper then introduces the Markov chain models, describes the instances analyzed, and presents the results of the study. Finally, this paper presents a motor design example before concluding with some insights gained from this study.

II. Related Work

A. Problem Solving

Designing involves satisfying a variety of constraints from users, regulators, and others, while making tradeoffs and compromises among competing objectives [7]. Empirical studies of real-world designers have described the problem-solving strategies that they use to generate solutions [8]. Although it may always be possible to redefine a design problem in terms of more fundamental considerations, in practice, someone determines the scope of the problem, and the design team must work within that formulation. For example, in the vehicle development process, the decision to launch the project requires the approval

Manuscript received July 15, 2014; revised January 5, 2015 and March 30, 2015; accepted July 13, 2015. Date of publication July 31, 2015; date of current version October 16, 2015. Review of this manuscript was arranged by Department Editor B. Jiang.

The author is with the Mechanical Engineering Department, University of Maryland, College Park, MD 20742 USA (e-mail: jwh2@umd.edu).

Digital Object Identifier 10.1109/TEM.2015.2456833

of multiple executives within the firm and a consensus about the new car's requirements, appearance, and financial viability [9]. At that point, the design problem that the vehicle development team must solve is fixed. In product development, the activities between concept generation and the delivery to market are the "problem-solving stage of development" [10]. The development of complex products requires solving a range of problems at different levels in the product hierarchy (from individual parts to major systems) [11]. Even relatively routine design tasks require problem solving [12].

March and Simon [2] described the general characteristics of human problem solving in organizational decision making. The first characteristic is that making a complex decision involves making a large number of small decisions. The second characteristic is that problem solving has a hierarchical structure in which solving any problem goes through phases that, in turn, require solving more detailed subproblems. The third characteristic is that problem solving consists of searching for possible solutions (cf., [13]). The fourth characteristic is that problem solving includes screening processes that evaluate the solutions that are found. The fifth characteristic is that problem solving has not only random components (such as finding and evaluating solutions) but also a procedural structure that allows it to yield good solutions. The proposed models of a bounded rational designer are motivated by these characteristics.

Designers search for ways to represent the problem, for solutions to the problem, and for information about the problem and the solutions. Simon [1] highlighted the important role that search plays in solving design problems. Based on their studies of human problem solving, Newell and Simon [14] described the problem-solving process as a search through a space of possible solutions and identified different types of solutions.

In some cases, designers search for a solution such as the product design. In other cases, they use conceptual and procedural knowledge to search for a set of operators or actions that will generate a solution (by transforming the initial state into a goal state) [15]. Within a product development project, problem solving requires searching through alternatives and evaluating them against the problem requirements and constraints [11]. For instance, a vehicle development team must find places inside a car to put many components and subassemblies (such as airbag crash sensors [16]).

Cognitive scientists have also described problem solving as search. Wang and Chiew [17] described human problem solving as a higher layer cognitive process and modeled the problem-solving process as a search that iteratively generates and evaluates potential solutions. The process also includes evaluating the acceptability of a solution and other lower layer cognitive functions. Payne and Wenger [18] identified the elements of problem solving as representation (which describes the possible states) and search (moving among those states).

Although design problem solving involves many types of activities, we conclude that search is a valid model of this process.

B. Subproblems

Alexander [19], Simon [1], and Lawson [7] described the hierarchical nature of design problems and noted that a design problem is a structured collection of subproblems. A complete design solution is the combination of solutions to multiple subproblems. Therefore, in the space of all possible solutions, there are some solutions that share the same solution to a specific subproblem. For instance, Simon [1] described a highway design problem, in which a "band of interest" is a set of specific routes that connect two cities. The design problem consisted of first selecting a band of interest and then selecting a specific route within that band of interest. All of the routes in the same band of interest form a distinct set. Engineering design processes often have a conceptual design process that is followed by a detailed design process. The detailed designs that elaborate on the same concept form a distinct set.

A separation is a process that solves a set of subproblems. A large problem is divided into subproblems, and the solution to one subproblem provides the inputs to one or more subsequent subproblems. Note that the separation does not have to be a simple sequence of subproblems; it may have subproblems that are solved in parallel at places. A given separation specifies a partial order in which the subproblems are solved. The quality of the solution generated depends upon the separation that is used [20], [21]. Design teams who inappropriately separate a design problem will generate low-quality solutions [22].

C. Modeling Teams

A great variety of models, including agent-based simulation models, have been used to study teams in general (cf., [23]). Teams of bounded rational decision-makers who perform the same task repeatedly were studied by Boettcher and Levis [24]–[26], who modeled the decision-making process of each person as a two-stage process that incorporated situation assessment and response selection. The condition of bounded rationality is modeled as a constraint on the rate at which each person can process information. The decision-making process was later extended to include information fusion, task processing, and command interpretation stages that were represented using Petri nets [27]. Sosa and Marle [28] presented a regression model for predicting the quality of solutions that are generated by a design team as a function of team characteristics, such as experience, familiarity, and team dynamics but did not consider the problem-solving process.

Gurnani and Lewis [29] studied collaborative decentralized design processes in which the models of the team's individual decision makers (the designers) were based on the ideas of bounded rationality. In their model, the value chosen by each designer was determined by randomly sampling from a distribution around the (locally) "optimal" solution. This model was meant to represent the mistakes that designers make due to bounded rationality. Their results showed that incorporating bounded rationality led to more desirable solutions in a collaborative decentralized design process in which the designers had different objectives and no way to coordinate their activities.

Herrmann [21] presented a method for assessing the quality of a product design process by measuring the profitability of the product that the process generates. This approach simulated the choices of a bounded rational designer for each subproblem using search algorithms. The searches, which were limited to a fixed number of iterations, had random components (either randomly selecting a solution or randomly moving to a point near the existing solution) and a procedural structure to keep track of the best solution found. The results showed that separating a problem into subproblems yields a better solution than solving the entire problem at once when bounded rational search is employed.

Hong and Page [30] modeled problem solvers of limited ability as searches that they called "heuristics," and each heuristic searched a finite set of solutions until it could not find a better solution. Thus, the problem solver is conducting a type of hill-climbing search. Hong and Page simulated teams of such problem solvers and identified conditions under which a diverse set of problem solvers will likely perform better than a team of high-performing individuals. In their work, the problem solvers searched a finite set of solutions (the size ranged from 200 to 10 000). A problem solver was characterized by how many and which points near the current solution it would consider. Essentially, different problem solvers had different neighborhood definitions. LiCalzi and Surucu [31] studied teams of problem solvers in which different problem solvers had different partitions of the solution space, which affected the team's ability to find the optimal solution. They considered how the size of the search space affected the size of the team needed to find the optimal solution.

D. Design Structure Matrix (DSM)

The DSM is an information-based model of product development. A DSM represents the activities in a product development project, their duration, and the probabilities of repeating them [32]–[34]. Recent work using DSMs has studied the impact of iterations and overlapping activities [35]–[37].

Unlike previous work that modeled product development using DSMs, this study focused on the quality of the design solution, not the time required to complete the task. Unlike Gurnani and Lewis [29], this study modeled a bounded rational decision maker with a search process. The Markov chain models used in this study are different from the simulation models used in previous work [21], [30], [31]. Moreover, this study, unlike Hong and Page [30] and LiCalzi and Surucu [31], did not measure the value of diversity; instead it studied how separating the problem and allocating the team's resources affect the quality of the solutions that teams of problem solvers generated.

III. Problem Formulation

This study considered teams of bounded rational decision makers (problem solvers or agents) and different approaches for solving a generic class of design problems, which allowed us to vary the characteristics of the search space directly without the distractions of specific decision variables, constraints, and objective functions. (To establish the approach more concretely, however, a specific example is discussed in Section VIII.)

In this paper, a design problem requires finding an optimal solution in a space of many solutions. The solution space is hierarchical, and it is divided into sets. Each set consists of similar solutions. (The sets are mutually exclusive and collectively exhaustive.)

In this study, every solution in the solution space has a value in the set $\{1, \ldots, N\}$, where the value N is the optimal value, and a solution with value i is better than a solution with value j if and only if $i > j$. The number of solutions in the solution space is not relevant. Let $p(x)$ be the probability that a randomly selected solution will have the value x, for all $x \in \{1, \ldots, N\}$.

Because of the hierarchical nature of the solution space, the solutions are grouped into sets. The value of a set depends upon the values of the solutions in the set. We assume that there is a single value in the range $\{1, \ldots, S\}$ that describes the value of a set. (Again, a larger value is better.) Let $q(s)$ be the probability, when searching for a set, that a randomly selected set will have the value s, for all $s = 1, \ldots, S$. Let $r_s(x)$ be the probability, when searching in a set with value s, that a randomly selected solution will have the value x. Naturally, the following relationship must hold: $p(x) = \sum_{s=1}^{S} q(s) r_s(x)$.

IV. Modeling Agents Searching

A problem-solver ("agent") who is searching a space of solutions (or sets) wants to find the best point (set), that is, the one that has the largest value. The agent is limited, however, by the limitations of the search strategy. In this study, we considered three basic models of an agent's search. We called these the "hill-climbing," "fixed effort," and "target values" models.

In all of the models, the agent begins by randomly selecting a solution in the solution space. From the current solution, the agent checks another randomly selected solution and accepts the new one if it is better (has a greater value) than the current one. The distinctive characteristic of each model was the stopping rule that determined when the agent stopped searching and returned the current solution (the best found so far). Note that the agent has a "memory" because it retains and returns the best solution that was found during the search.

The search for a set works in the same way. There are S types of sets and an ordering over the values of the sets such that a set of type a is better than a set of type b if $a > b$. An agent who is searching for a set returns the best set found.

In the "hill-climbing" model, the agent's search is governed by a parameter K. If the agent checks K consecutive solutions that are not better than the current one, then the agent stops his search. (Increasing K will increase the effort spent searching and will increase the expected value of the solution found.)

The quality of the solution found by the agent is a random variable that depends upon K and the distribution of values in the search space. Let $v(t)$ be the value of the agent's current point after checking t points (where $t = 0$ corresponds to the initial point).

The stochastic process $v(t)$, $t = 1, 2, \ldots$, is generated by a Markov chain. In the hill-climbing model, the state of the Markov chain is the value of the agent's current point and the number of points checked since accepting that point. Let $(v(t), n(t))$ be the state at step t, where $v(t)$ is the value of the agent's current point and $n(t)$ equals the number of points checked since accepting that point. If a new point is not better, and, thus, not accepted, $v(t)$ remains the same, and $n(t)$ increases. If the new point is better, however, $v(t)$ becomes the value of the new point, and $n(t)$ becomes 0 again.

The number of states is finite. The states are the elements of $\{1,\ldots,N\} \times \{0,\ldots,K\}$. At $t = 0$, $n(0) = 0$, while $P\{v(0) = x\}$ is determined by the initial distribution. (In some cases, $P\{v(0) = x\} = p(x)$.) All states that have $n(t) = K$ are absorbing states because the search stops when this occurs. All of the other states are transient. Thus, the distribution of $v(t)$ converges to the distribution over the absorbing states (1, *K*), ..., (*N*, *K*). Let $p'(x)$ be the probability that the search stops at state (*x*, *K*), and note that this distribution depends upon the initial distribution over the states.

The probability transition matrix $P = \Phi(p)$ can be described as follows:

$$\begin{aligned}
\text{Let } P(w,j,x,i) &= P\big\{v(t) = x, n(t) = i | v(t-1) \\
&= w, n(t-1) = j\big\} \\
P(w,j,x,0) &= p(x), \text{ for } w = 1,\ldots,N-1,\ x \\
&= w+1,\ldots,N,\ j = 0,\ldots,K-1 \\
P(w,j,w,j+1) &= p(1) + \cdots + p(w), \text{ for } w \\
&= 1,\ldots,N, j = 0,\ldots,K-1 \\
P(w,K,w,K) &= 1, \text{for } w = 1,\ldots,N \\
P(w,j,x,i) &= 0,\ \text{otherwise.}
\end{aligned}$$

In the "fixed effort" and "target values" models, the state is only $v(t)$, the value of the best solution found so far. In the "fixed effort" model, the agent checks a fixed number of randomly selected solutions. The probability transition matrix $F = \theta(p)$ can be described as follows:

$$\begin{aligned}
&\text{Let } F(w,x) = P\{v(t) = x | v(t-1) = w\} \\
&F(w,x) = p(x), \text{for } w = 1,\ldots,N-1, x = w+1,\ldots,N \\
&F(w,w) = p(1) + \cdots + p(w), \text{for } w = 1,\ldots,N \\
&F(w,x) = 0, \text{for } w = 2,\ldots,N, x = 1,\ldots,w-1.
\end{aligned}$$

The only absorbing state is *N*. The result of conducting a search with *m* steps is $p' = pP^m$.

In the "target values" model, the agent searches until a sufficiently good (near-optimal) solution is found. Let *T* denotes the set of target values. Let n_v be the number of target values. Set $V = N + 1 - n_v$. Then, $T = \{V, V+1,\ldots,N\}$. The probability transition matrix $U = \Upsilon(p)$ can be described as follows:

$$\begin{aligned}
&\text{Let } U(w,x) = P\{v(t) = x | v(t-1) = w\} \\
&U(w,w) = p(1) + \cdots + p(w), \text{for } w = 1,\ldots,V-1 \\
&U(w,x) = p(x), \text{for } w = 1,\ldots,V-1, x = w+1,\ldots,N \\
&U(w,w) = 1, \text{for } w = V,\ldots,N \\
&U(w,x) = 0 \, \text{otherwise.}
\end{aligned}$$

The target values $\{V, V+1,\ldots,N\}$ are absorbing states. Let $p' = \tau(p, n_v)$ be the probability distribution over the target values (at the end of the search) as a function of the distribution of the values in the search space and the number of target values. Then, $p'(x) = p(x) / \sum_{v \in T} p(v)$, for $x \in T$.

We considered two approaches for modeling collaboration (when two or more agents work together to solve the same problem). In the first approach, which we call "serial," a collaborating agent waits for the first agent to finish a search and continues the search where the first one ended. In this case, the probability distribution over the second agent's initial state equals the probability distribution over the first agent's final state. If a third agent continues the search, then the probability distribution over the third agent's initial state equals the probability distribution over the second agent's final state.

In the second approach, which we call "united," the collaborating agents search at the same time. At each step, the collaborating agents select independent points in the solution space and keep the best of these points; if the value of this best point is better than the current point, then they accept the new point. Although the structure of the probability transition matrix remains the same, the transition probabilities are different because the distribution of the best point is different from the distribution of a single point.

If two or three agents are united, then the probability that the best of their values (on any step) can be determined as follows. Let $c(x)$ be the probability that an agent selects a value less than *x*. $c(1) = 0$, and $c(x) = \sum_{y=1}^{x-1} p(y)$, for $x = 2,\ldots,N$.

Let $\tilde{p} = \Gamma(p)$ be the probability that the best value selected by two collaborating agents equals *x*, and let $\hat{p} = \Delta(p)$ be the probability that the best value selected by three collaborating agents equals *x* $\tilde{p}(x) = p^2(x) + 2p(x)c(x)$, and $\hat{p}(x) = (p(x) + c(x))^3 - c^3(x)$.

A team of two agents can solve a problem using two approaches. In the "all-at-once" approach, both agents collaborate to search the entire space of solutions. The agents can also separate the problem into two subproblems that are solved sequentially. The first subproblem is to find the best set of solutions, and the second subproblem is to find the best solution within that set. The first agent searches the sets until he stops, and then, the second agent searches that set of solutions until he stops. (We will refer to this as a "1-1 separation").

If the team consists of three agents, then there are three approaches.

1) All three collaborate to search the entire space of solutions.
2) Two agents collaborate to search the sets, and then, the third agent searches that set of points (a "2-1" separation).
3) One agent searches the sets, and then, the other two agents collaborate to search that set of points (a "1-2 separation").

Although our primary measure of effort is the number of people on the team solving the design problem, we can exploit these models to determine the expected number of steps taken in each search by determining the first passage time to an absorbing state (see, for instance, Bhat [38]). (Of course, the fixed effort searches have a given number of steps.)

V. Instances

To compare the performance of different separations, we generated different distributions for the sets and populations to generate nine cases. To consider cases that had a range of difficulty, we varied the difficulty of finding high-value sets and the difficulty of finding high-value points within these sets. In the "easiest" cases, every value was equally (or nearly equally) likely. In the more difficult cases, the high-value sets and points were less likely. The distributions chosen were consistent with these principles. Note that, in the experiments described by Hong and Page [30], all of the possible values were equally likely.

Each case is distinguished by a specific distribution for $q(s)$, the probability of finding a type s set when searching for a set, and specific distributions for $r_s(x)$, the probability of finding a point with value x when searching in a type s set. (Together, these determine $p(x)$.)

We generated nine cases by combining three distributions for $q(s)$ (which we will call A, B, and C) and three sets of distributions for $r_s(x)$ (which we will call I, II, and III). Thus, we have nine cases, labeled from A-I to C-III. All of the cases have a space with $S = 15$ types of sets and $N = 120$ values.

In the A cases, all of the sets were equally likely $q_A(s) = \frac{1}{15}$ for all $s = 1, \ldots, S$. In the B cases, the higher value sets were less likely (the overall distribution is triangular): $q_B(s) = (16 - s)/120$ for all s. In the C cases, the higher value sets were even less likely (the overall distribution is geometric) $q_C(s) = 0.7^{s-1}/(1 + \cdots + 0.7^{14}) = 0.3(0.7)^{s-1}/(1 - 0.7^{15})$, for all s.

In the type-I cases, within a set, every value was possible, but some values were slightly more likely. In particular, for $s = 1, \ldots, 15$, $r_s(x) = \frac{19}{2000} = 0.0095$, for $x = 5(s-1) + 1, \ldots, 5(s-1) + 50$ and $\frac{15}{2000} = 0.0075$, otherwise.

In the type-II cases, within a set, only some values were possible, but the possible values were equally likely. In particular, for $s = 1, \ldots, 15$, $r_s(x) = \frac{1}{50}$, for $x = 5(s-1) + 1, \ldots, 5(s-1) + 50$ and 0, otherwise.

In the type-III cases, within a set, only some values were possible, but the greater values were less likely. In particular, for $s = 1, \ldots, 15$, $r_s(x) = (5s + 46 - x)/1275$, for $x = 5(s-1) + 1, \ldots, 5(s-1) + 50$ and 0, otherwise.

Note that, in the type-II and type-III cases, only the best sets have optimal values.

VI. Computational Experiments

The purpose of the computational experiments was to compare the performance of the teams' solution approaches. We compared the results of the 1-1 separation to the results of two agents conducting an all-at-once search. We compared the results of the 1-2 separation and the 2-1 separation to the results of three agents conducting an all-at-once search.

Each search was modeled using the hill-climbing stopping rule (with $K = 1, 2, 3, 4$), the fixed effort stopping rule (with $m = 10, 20, 30,$ and 40 steps per search), and the target values stopping rule (with $n_v = 1$, 2, 3, and 4 target values).

When using the hill-climbing stopping rule, the collaboration of multiple agents was modeled as a serial search and as a united search. When using the fixed effort stopping rule, only the serial search was considered because the model with the united search is equivalent. (In the united search, each step returns the best value found by multiple agents, which is equivalent to every agent taking one step and returning one value.) When using the target values stopping rule, only the united search was considered. A serial version of collaboration would not make sense because the first agent searches until a target value is found.

In addition, we used the target values stopping rule only when considering the type-I cases because every set contained every target value (in the type-II and type-III cases, some sets contained none of the target values).

For each combination of case, search (all-at-once or separation), stopping rule, and stopping rule parameter value, we determined the probability distribution of the value returned at the end of the search and the expected number of steps required. Due to space limitations, we cannot provide the specific equations used for every calculation. Here, we will briefly outline the computational approach.

For an all-at-once search modeled with a serial team, we determined $p' = \Omega(p, P)$, the probability distribution over the values that the first agent finds; then, $p'' = \Omega(p', P)$, the probability distribution over the values that the second agent finds; and finally $p'' = \Omega(p', P)$, the probability distribution over the values that the third agent finds. (The technique for calculating these depends upon the stopping rule being used.)

For a united team of two agents, we determined the distribution $\tilde{p} = \Gamma(p)$, the transition probability matrix $\tilde{P} = \Phi(\tilde{p})$, and the result $\tilde{p}' = \Omega(\tilde{p}, \tilde{P})$, the probability distribution over the values that the team finds. For a united team of three agents, we determined the distribution $\hat{p} = \Delta(p)$, the transition probability matrix $\hat{P} = \Phi(\hat{p})$, and the result $\hat{p}' = \Omega(\hat{p}, \hat{P})$, the probability distribution over the values that the team finds.

For the separations, we had to determine the results of searching for the best set and the results of searching for the best solution in each set.

If two agents separate the problem, and each one solves one subproblem (a "1-1 separation"), then we determined the transition probability matrix $Q = \Phi(q)$ and the result $q' = \Omega(q, Q)$ for the first agent. For every set, we determined the transition probability matrix $R_s = \Phi(r_s)$ and the result $r'_s = \Omega(r_s, R_s)$, and then, used these results to determine $p^{(11)}(x) = \sum_{s=1}^{S} q'(s) r'_s(x)$.

If three agents separate the problem as a "1-2 separation," then $q' = \Omega(q, Q)$ is the result for the first agent. The other two agents collaborate to find a solution.

For a serial team, then, for each set, $r'_s = \Omega(r_s, R_s)$ is the result for the second agent (the first for this subproblem). Then, $r''_s = \Omega(r'_s, R_s)$ is the result for the last agent (the second for this subproblem). We used these results to determine the team result $p^{(12)}(x) = \sum_{s=1}^{S} q'(s) r''_s(x)$.

For a united team, for every set, we determined the probability distribution $\tilde{r}_s = \Gamma(r_s)$, the transition probability matrix $\tilde{R}_s = \Phi(\tilde{r}_s)$, and the result $\tilde{r}'_s = \Omega(\tilde{r}_s, \tilde{R}_s)$. We used these results to determine the team result $\tilde{p}^{(12)}(x) = \sum_{s=1}^{S} q'(s) \tilde{r}'_s(x)$.

If the agents separate the problem as a "2-1 separation," then the first two agents collaborate to search for a set, and the third agent searches for a solution.

TABLE I
LIKELIHOOD OF AN OPTIMAL SOLUTION IN EACH CASE

Case	p(120)
A-I	7.6×10^{-3}
A-II	1.3×10^{-3}
A-III	5.2×10^{-5}
B-I	7.5×10^{-3}
B-II	1.7×10^{-4}
B-III	6.5×10^{-6}
C-I	7.5×10^{-3}
C-II	4.1×10^{-5}
C-III	1.6×10^{-6}

For a serial team, the second agent continues where the first agent stopped, so the result $q'' = \Omega(q', Q)$. We used these results to determine the team result $p^{(21)}(x) = \sum_{s=1}^{S} q''(s) r'_s(x)$.

For a united team, we determined the probability distribution $\tilde{q} = \Gamma(q)$, the transition probability matrix $\tilde{Q} = \Phi(\tilde{q})$, and the result $\tilde{q}' = \Omega(\tilde{q}, \tilde{Q})$. We used these results to determine the team result $\tilde{p}^{(21)}(x) = \sum_{s=1}^{S} \tilde{q}'(s) r'_s(x)$.

When calculating the expected number of steps in a search, the expected number of steps for unified team searches was multiplied by the number of agents in the team.

VII. RESULTS

Here, we report the probability that an agent (or team) will find a solution with the optimal value, and the expected number of steps that an agent (or team) spends searching. Because the primary purpose is to compare the separations to the all-at-once approaches, the results for the 1-1 separation are shown relative to the results for the two-agent team performing an all-at-once search. Moreover, the results for the 1-2 separation and the 2-1 separation are shown relative to the results for the three-agent team performing an all-at-once search. Because varying the search parameter did not affect the relative performance of the separations, all of the results reported here are averaged over the four values of *K*, the four values of effort *m*, or the four values of n_v, the number of target values.

Although the absolute performance of the 1-1 separation does not depend on the team collaboration because each search is done by only one agent, the absolute performance of the all-at-once search with two agents does depend on the team collaboration, which affects the relative performance of the 1-1 separation.

The likelihood of an optimal solution (in the entire population) varied from case to case as shown in Table I. These values show that cases A-I, B-I, and C-I are easier because optimal solutions are more likely.

The absolute performance of the models did depend upon the parameters *K*, *m*, and n_v. For example, we will consider the performance of the three-agent all-at-once search on case B-I. The hill-climbing model with serial teams predicted that, as *K* increased from 1 to 4, the likelihood of finding an optimal solution increased from 0.039 to 0.125 and the expected number of steps increased from 4.3 to 16.7. The hill-climbing model with a unified team predicted that, as *K* increased from 1 to 4, the likelihood of finding an optimal solution increased from 0.055 to 0.161 and the expected number of steps increased from 6.5 to 22.4. The fixed effort model predicted that, as the number of steps increased from 30 to 120, the likelihood of finding an optimal solution increased from 0.209 to 0.599. The target values model predicted that, as the number of target values increased from 1 to 4, the likelihood of finding an optimal solution decreased from 1 to 0.256 and the expected number of steps decreased from 134.0 to 34.3.

TABLE II
AVERAGE RELATIVE PROBABILITY THAT A TEAM OF TWO AGENTS WILL FIND AN OPTIMAL SOLUTION USING A 1-1 SEPARATION FOR DIFFERENT SEARCH MODELS

Case	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)	Target values (united)
A-I	0.675	0.558	0.653	0.994
A-II	2.535	2.082	4.808	
A-III	2.660	2.181	5.927	
B-I	0.645	0.533	0.584	0.994
B-II	2.938	2.409	8.977	
B-III	3.103	2.544	11.621	
C-I	0.641	0.529	0.567	0.994
C-II	2.926	2.400	9.761	
C-III	3.092	2.535	12.715	

TABLE III
AVERAGE RELATIVE PROBABILITY THAT A TEAM OF THREE AGENTS WILL FIND AN OPTIMAL SOLUTION USING A 1-2 SEPARATION FOR DIFFERENT SEARCH MODELS

Case	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)	Target values (united)
A-I	0.796	0.720	0.839	0.995
A-II	2.915	2.540	5.150	
A-III	3.160	2.828	7.711	
B-I	0.761	0.689	0.760	0.995
B-II	3.373	2.926	9.335	
B-III	3.688	3.300	15.119	
C-I	0.757	0.685	0.740	0.995
C-II	3.357	2.911	10.111	
C-III	3.674	3.287	16.542	

The relative performance of the 1-1 separation varied across the cases. As shown in Table II, in cases A-I, B-I, and C-I, using the separation decreased the likelihood of returning an optimal solution. In the other cases, using the separation increased the likelihood of returning an optimal solution. These results were consistent across the different models. As shown in Table V, the change in the expected number of steps in a search varied; different models produced different results. The hill-climbing model with serial collaboration predicted that using the separation would increase the expected number of steps, but the hill-climbing model with united teams predicted that using the separation would decrease the expected number of steps. Of course, the fixed effort predicted no change, but the target values model predicted an increase that varied based on the distribution of the sets.

TABLE IV
AVERAGE RELATIVE PROBABILITY THAT A TEAM OF THREE AGENTS WILL FIND AN OPTIMAL SOLUTION USING A 2-1 SEPARATION FOR DIFFERENT SEARCH MODELS

Case	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)	Target values (united)
A-I	0.525	0.397	0.494	0.989
A-II	2.723	2.238	3.925	
A-III	2.851	2.333	4.719	
B-I	0.490	0.367	0.443	0.989
B-II	3.460	3.028	10.601	
B-III	3.654	3.197	13.653	
C-I	0.485	0.363	0.420	0.989
C-II	3.515	3.116	12.482	
C-III	3.715	3.293	16.244	

TABLE V
AVERAGE RELATIVE NUMBER OF STEPS REQUIRED BY A TEAM OF TWO AGENTS USING A 1-1 SEPARATION FOR DIFFERENT SEARCH MODELS

Case	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)	Target values (united)
A-I	1.113	0.884	1	1.011
A-II	1.107	0.878	1	
A-III	1.108	0.879	1	
B-I	1.114	0.885	1	1.483
B-II	1.109	0.879	1	
B-III	1.110	0.880	1	
C-I	1.096	0.870	1	3.661
C-II	1.092	0.867	1	
C-III	1.093	0.867	1	

TABLE VI
AVERAGE RELATIVE NUMBER OF STEPS REQUIRED BY A TEAM OF THREE AGENTS USING A 1-2 SEPARATION FOR DIFFERENT SEARCH MODELS

Case	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)	Target values (united)
A-I	1.105	0.916	1	1.010
A-II	1.098	0.903	1	
A-III	1.100	0.909	1	
B-I	1.106	0.917	1	1.479
B-II	1.099	0.903	1	
B-III	1.102	0.909	1	
C-I	1.093	0.908	1	3.639
C-II	1.087	0.895	1	
C-III	1.090	0.901	1	

The results for the relative performance of the 1-2 separation and the 2-1 separation varied across the cases in a similar way as shown in Tables III, IV, VI, and VII. Again, in cases A-I, B-I, and C-I, using the separation decreased the likelihood of returning an optimal solution, but, in the other cases, using the separation increased the likelihood of returning an optimal solution. The results for the expected number of steps showed similar trends as those seen for the relative performance of the 1-1 separation.

The performance of the 1-2 separation and the 2-1 separation was similar but not identical. As shown in Tables VI and VII, the hill-climbing models determined that the 1-2 separation would require more steps than the 2-1 separation. Table VIII shows that the hill-climbing and fixed effort models determined that the 2-1 separation would have a higher likelihood of returning an optimal solution than the 1-2 separation in some of the cases. The fixed effort model predicted larger differences between the two separations than the hill-climbing model.

TABLE VII
AVERAGE RELATIVE NUMBER OF STEPS REQUIRED BY A TEAM OF THREE AGENTS USING A 2-1 SEPARATION FOR DIFFERENT SEARCH MODELS

Case	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)	Target values (united)
A-I	1.092	0.884	1	1.008
A-II	1.088	0.878	1	
A-III	1.088	0.879	1	
B-I	1.099	0.900	1	1.479
B-II	1.095	0.893	1	
B-III	1.095	0.894	1	
C-I	1.086	0.890	1	3.639
C-II	1.083	0.885	1	
C-III	1.083	0.885	1	

TABLE VIII
DIFFERENCE IN THE AVERAGE RELATIVE PROBABILITY THAT A TEAM OF THREE AGENTS USING A 2-1 SEPARATION AND A TEAM USING A 1-2 SEPARATION WILL FIND AN OPTIMAL SOLUTION FOR DIFFERENT SEARCH MODELS

Case	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)	Target values (united)
A-I	−0.271	−0.323	−0.346	−0.006
A-II	−0.192	−0.302	−1.225	
A-III	−0.309	−0.495	−2.992	
B-I	−0.271	−0.321	−0.317	−0.006
B-II	0.087	0.103	1.266	
B-III	−0.034	−0.103	−1.465	
C-I	−0.272	−0.322	−0.320	−0.006
C-II	0.158	0.206	2.371	
C-III	0.041	0.006	−0.298	

VIII. MOTOR DESIGN

This section describes a specific design problem and the results of testing the different separations on this problem. Consider a design team who is designing a transverse flux linear motor (TFLM). The design team must determine values for four design variables (described below) and seeks to maximize maximum thrust force Y_T, minimize maximum detent force Y_D, and minimize weight Y_W.

To evaluate the team's performance using different separations of this problem, we first use a TFLM model to determine the team's search space. Note that the team is not explicitly formulating and solving this TFLM model. We assume that the design team is searching for a solution in a manner consistent with the Markov chain models presented earlier.

In the TFLM model [39], the four design variables X_1, X_2, X_3, X_4 are the stator pole length (mm), the air gap length (mm), the winding window width (mm), and the stator pole width (mm). These design variables are constrained to the following bounds:

$$10 \leq X_1 \leq 30$$
$$0.5 \leq X_2 \leq 1.5$$
$$20 \leq X_3 \leq 40$$
$$10 \leq X_4 \leq 18.$$

After conducting a FEA on 31 design points, Hasanien developed response surface models for the objectives Y_T, Y_D, and Y_W. In these models, each design variable X_i was represented by a scaled variable x_i. The range of each scaled variable was [−1, 1]. The scaling was linear $x_i = -1$ when X_i was at the minimum of its feasible range, $x_i = 0$ when X_i was at the midpoint of its feasible range, and $x_i = 1$ when X_i was at the maximum of its feasible range. The second-order polynomial functions for each objective were given in terms of these scaled variables

$$\begin{aligned} Y_W = \; & 10.09 + 1.86x_1 - 0.020x_2 + 2.39x_3 + 0.58x_4 \\ & + 0.092x_1^2 + 0.097x_2^2 - 0.327x_3^2 + 0.177x_4^2 \\ & - 0.0006x_1x_2 + 0.0556x_1x_3 + 0.258x_1x_4 \\ & - 0.0006x_2x_3 - 0.0006x_2x_4 + 0.0206x_3x_4 \end{aligned}$$

$$\begin{aligned} Y_T = \; & 707.35 + 118.08x_1 - 88.03x_2 + 36.61x_3 + 44.95x_4 \\ & - 71.43x_1^2 + 0.746x_2^2 - 17.78x_3^2 - 67.83x_4^2 \\ & - 8.57x_1x_2 - 14.68x_1x_3 - 9.00x_1x_4 \\ & + 7.46x_2x_3 + 10.99x_2x_4 + 12.15x_3x_4 \end{aligned}$$

$$\begin{aligned} Y_W = \; & 74.78 + 11.58x_1 - 9.38x_2 + 13.51x_3 + 12.84x_4 \\ & - 14.35x_1^2 + 3.24x_2^2 - 1.17x_3^2 + 1.67x_4^2 \\ & - 4.45x_1x_2 - 2.49x_1x_3 + 1.64x_1x_4 \\ & - 1.11x_2x_3 - 2.73x_2x_4 + 0.19x_3x_4. \end{aligned}$$

Based on these variables, we developed a multiattribute utility function. For each attribute (weight, thrust force, and detent force), the utility function was linear over a range from the original values and the best reported value [39]

$$U_W = (10.09 - Y_W)/(10.09 - 5.60)$$
$$U_T = (Y_T - 709.9)/(804.65 - 709.9)$$
$$U_D = (74.61 - Y_D)/(74.61 - 26.71).$$

The total utility was a linear combination of these utility values; the weights were chosen so that the optimal solution to the resulting problem was near the near-optimal solutions given by Hasanien [39]: $U = 0.4U_W + 0.2U_T + 0.4U_D$.

In this problem, a set represented a portion of the solution space. In particular, the range of each scaled design variable was divided into two intervals [−1, 0] and [0, 1]. Each of the 16 sets was a different combination of these intervals. For example, one set was the Cartesian product $\{-1 \leq x_1 \leq 0\} \times \{-1 \leq x_2 \leq 0\} \times \{-1 \leq x_3 \leq 0\} \times \{-1 \leq x_4 \leq 0\}$.

TABLE IX
AVERAGE RELATIVE LIKELIHOOD OF FINDING AN OPTIMAL SOLUTION BY A TEAM OF AGENTS FOR DIFFERENT SEARCH MODELS

Separation	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)
1-1	2.21	1.87	2.74
1-2	2.50	2.17	2.63
2-1	2.17	1.80	2.10

All of the sets were considered equally likely, so $q(s) = 1/16$ for all 16 sets. The "value" of a set was determined by evaluating the utility of the solution in the middle of the set (every scaled variable was at the midpoint of its interval). For example, the value of the set mentioned in the previous paragraph was determined by the utility of the point (−0.5, −0.5, −0.5, −0.5).

For purposes of constructing the Markov chains that represent the design team's searches, the utility values of the solutions were put into "bins." There were 40 equal-size bins that spanned the range from −1 to 1. Each bin corresponded to one value in the Markov chain. A larger utility is preferred.

Within each set, we divided the interval for each scaled design variable into ten equal subintervals and found the midpoints of each subinterval, which yielded ten values for this scaled design variable and 10^4 solutions, all in the specified set. We created the distribution $r_s(v)$ for the set by evaluating the utility of all 10^4 solutions and determining the frequency of each bin. The distribution within each set was roughly triangular.

The 160 000 solutions yielded utilities in only 19 distinct bins. Only 721 solutions (from four sets) had utilities in the optimal bin. Thus, the fraction of all solutions that are optimal is approximately 0.0045.

Using this instance, we tested the same all-at-once searches and the separations that were used in the computational experiment. The hill-climbing model was run with $K = 1, 2, 3, 4$, and 5. The fixed effort model was run with 10, 20, 30, 40, and 50 steps per search. (Because not every set had an optimal solution, the target values model was not used for this example.)

As we did in the computational experiments, we determined the performance of the 1-1 separation relative to the results for the two-agent team performing an all-at-once search and the performance of the 1-2 separation and the 2-1 separation relative to the results for the three-agent team performing an all-at-once search. Because varying the model parameters did not affect the relative performance of the results, all of the results reported here are averaged over the five values of *K* and the five values of effort *m*.

As shown in Table IX, using separations increased the likelihood that the design team finds an optimal solution. This result was consistent across the different models. The average relative likelihood ranged from 1.80 to 2.74.

As shown in Table X, the change in the expected number of steps in a search varied, and different models produced different results. The hill-climbing model with serial collaboration predicted that using the separation would increase the expected number of steps, but the hill-climbing model with united teams

TABLE X
AVERAGE RELATIVE NUMBER OF STEPS REQUIRED BY A TEAM OF AGENTS FOR DIFFERENT SEARCH MODELS

Separation	Hill climbing (serial)	Hill climbing (united)	Fixed effort (serial)
1-1	1.11	0.90	1
1-2	1.09	0.91	1
2-1	1.10	0.92	1

predicted that using the separation would decrease the expected number of steps. Of course, the fixed effort model predicted no change.

The performance of the two separations with three agents (the 1-2 and 2-1 separations) was the same on one measure (the average relative number of steps required) but different on the other measure (the average relative likelihood of finding an optimal solution). On the latter measure, the 1-2 separation performed better than the 2-1 separation. The sets were equally likely, so it was not difficult to find a set with high-quality solutions; within a set, however, the optimal solutions were less likely. Thus, spending more effort searching the set (as the 1-2 separation does) is more effective than spending more effort for a set with high-quality solutions.

These results indicate that, for this problem, using a separation increases the likelihood that the design team finds an optimal solution (compared to all-at-once searches), and a third agent, if available, should search for solutions (not sets).

IX. SUMMARY AND CONCLUSION

This paper presented Markov chain models for evaluating the search of a bounded rational decision maker and discussed the results of a study that analyzed the relative performance of teams who are solving design problems. This study was motivated by the following questions.

1) When is separating a problem a better approach than solving the problem all-at-once?
2) What is the best way to assign members of the team to the subproblems in a separation?

This paper's main contribution is a novel quantitative approach for answering these two questions.

The results indicate that there are some situations in which searching the complete space of solutions is a better approach and other situations in which separating the problem is a better approach. In particular, the separation approaches performed better than the all-at-once approaches when high-value solutions are less likely. The best assignment of team members (agents) to subproblems also varied. When finding good sets is difficult, more resources should be devoted to this task; otherwise, when finding good solutions within a set is difficult, more resources should be devoted to that subproblem.

In engineering design, these results suggest that more effort should be spent developing better concepts in those domains in which a variety of concepts exists but high-quality concepts are hard to find. In domains, in which the concepts all have similar performance, more effort should be spent searching for better detailed designs that implement the selected concept.

Beyond the particular results, the approach presented here is a novel quantitative approach for answering these questions. Although it requires some calculations, it does not require repeated simulation runs.

Applying this approach, for a class of design problems, would require identifying the overall design problem and the potential separations of that problem into subproblems (cf., [21]). For each problem and subproblem, the probability distribution over the quality of its feasible solutions needs to be determined; this could be done by creating a mathematical or computation model that predicts the solution quality as a function of the design variables, generating and evaluating feasible solutions, and inferring the distribution from these points. Then, the Markov chains that represent the searches for solutions can be constructed and analyzed as discussed in this paper.

The primary implication for those who design and manage design processes and engineering teams is that the quality of the solutions that the teams generate depends upon the separation that is being used (in addition to many other factors). Thus, improving a design process so that it generates better solutions may require redefining the subproblems that the teams are solving (cf., [40]).

This study, like the one described by Hong and Page [30], considered a class of generic problems. Doing this allowed us to vary the characteristics of the search space directly without the distractions of specific decision variables, constraints, and objective functions.

In this study, every agent who was searching the same space was modeled with the same probability distribution over the value of the next point tried, but this approach could be modified easily to model agents with different skills who sample the search space in different ways.

Trying to consider all of the search space at once can lead to design problems that are too large to solve well because the distribution of high-quality points makes it difficult for the bounded rational decision maker to approach an optimal solution. In these situations, using well-designed separations can be the best way to find quality solutions and make engineering design decisions. A separation is effective because human decision makers have limitations and cannot optimize.

Although this study did not consider the process of problem formulation, using a separation involves a type of problem formulation. In the separations that were studied, the solution to the first subproblem (finding a set) defined the search space for the second subproblem (finding a solution in that set).

These results reinforce the conclusions of Herrmann [21] about the usefulness of separating complex design problems for bounded rational decision makers. They also demonstrate the usefulness of this analysis approach for comparing separations that have different subproblems and different allocations of team resources.

REFERENCES

[1] H. A. Simon, *Models of Bounded Rationality*, vol. 3. Cambridge, MA, USA: MIT Press, 1997.
[2] J. March and H. Simon, *Organizations*, 2nd ed. Boston, MA, USA: Blackwell, 1993.

[3] H. A. Simon, *Administrative Behavior*, 4th ed. New York, NY, USA: Free Press, 1997.

[4] G. Gigerenzer, P. M. Todd, and ABC Research Group, *Simple Heuristics That Make us Smart*. New York, NY, USA: Oxford Univ. Press, 1999.

[5] G. Klein, J. K. Phillips, E. L. Rall, and D. A. Peluso, “A data-frame theory of sensemaking,” in *Expertise Out of Context: Proceedings of the Sixth International Conference on Naturalistic Decision Making*, R. R. Hoffman Ed. New York, NY, USA: Lawrence Erlbaum, 2007, ch. 6, pp. 113–155.

[6] K. E. Weick, *Sensemaking in Organizations*, Newbury Park, CA, USA: Sage, 1995.

[7] B. Lawson, *How Designers Think: The Design Process Demystified*. 3rd ed. Oxford, U.K.: Architectural Press, 1997.

[8] L. J. Ball, J. St, B. T. Evans, and I. Dennis, “Cognitive processes in engineering design: A longitudinal study,” *Ergonomics*, vol. 37, no. 11, pp. 1753–1786, 1994.

[9] J. A. Donndelinger, “A decision-based perspective on the vehicle development process,” in *Decision Making in Engineering Design*, K. Lewis, W. Chen, and L. C. Schmidt, Eds. New York, NY, USA: ASME, 2006, ch. 19.

[10] S. Myers and D. G. Marquis, *Successful Industrial Innovations: A Study of factors Underlying Innovation in Selected Firms*. Washington, DC, USA: National Science Foundation, 1969.

[11] S. Thomke and T. Fujimoto, “The effect of front-loading problem-solving on product development performance,” *J. Prod. Innov. Manage.*, vol. 17, pp. 128–142, 2000.

[12] D. C. Brown and B. Chandrasekaran, *Design Problem Solving: Knowledge Structures and Control Strategies*. San Mateo, CA, USA: Morgan Kaufmann, 1989.

[13] P. C. Nutt, “Search during decision-making,” *Eur. J. Oper. Res.*, vol. 160, pp. 851–876, 2005.

[14] A. Newell and H. A. Simon, *Human Problem Solving*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1972.

[15] M. Hegarty, “Knowledge and processes in mechanical problem solving,” in *Complex Problem Solving: Principles and Mechanisms*, Robert J. Sternberg and Peter A. Frensch, Eds. Hillsdale, NJ, USA: L. Erlbaum Associates, 1991.

[16] M. Walton, *Car: A Drama of the American Workplace*. New York, NY, USA: Norton, 1997.

[17] Y. Wang and V. Chiew, “On the cognitive process of human problem solving,” *Cogn. Syst. Res.*, vol. 11, no. 1, pp. 81–92, 2010.

[18] D. G. Payne and M. J. Wenger, *Cognitive Psychology*. Boston, MA, USA: Houghton Mifflin, 1998.

[19] C. Alexander, *Notes on the Synthesis of Form*, vol. 5. Cambridge, MA, USA: Harvard Univ. Press, 1964.

[20] P. Karimian and J. W. Herrmann, “Separating design optimization problems into decision-based design processes,” *J. Mech. Des.*, vol. 131, no. 1, p. 011007, 2009.

[21] J. W. Herrmann, “Progressive design processes and bounded rational designers,” *J. Mech. Des.*, vol. 132, no. 8, p. 081005, Aug. 2010.

[22] J. Austin-Breneman, T. Honda, and M. C. Yang, “A study of student design team behaviors in complex system design,” *J. Mech. Des.*, vol. 134, no. 12, pp. 124504-1–124504-7, 2012.

[23] R. M. Crowder, M. A. Robinson, H. P. N. Hughes, and Y.-W. Sim, “The development of an agent-based modeling framework for simulating engineering team work,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 6, pp. 1425–1439, Nov. 2012.

[24] K. L. Boettcher and A. H. Levis, “Modeling the interacting decisionmaker with bounded rationality,” *IEEE Trans. Syst. Man Cybern.*, vol. SMC-12, no. 3, pp. 334–344, May 1982.

[25] K. L. Boettcher and A. H. Levis, “Modeling and analysis of teams of interacting decisionmakers with bounded rationality,” *Automatica*, vol. 19, no. 6, pp. 703–709, 1983.

[26] A. H. Levis and K. L. Boettcher, “Decisionmaking organizations with acyclical information structures,” *IEEE Trans. Syst. Man Cybern.*, vol. SMC-13, no. 3, pp. 384–391, Jun. 1983.

[27] A. H. Levis, “Executable models of decision making organizations,” in *Organizational Simulation*, W. B. Rouse and K. R. Boff, Eds. Hoboken, NJ, USA: Wiley, 2005.

[28] M. E. Sosa and F. Marle, “Assembling creative teams in new product development using creative team familiarity,” *J. Mech. Des.*, vol. 135, no. 8, pp. 081009-1–081009-13, 2013.

[29] A. Gurnani and K. Lewis, “Collaborative, decentralized engineering design at the edge of rationality,” *J. Mech. Des.*, vol. 130, no. 12, p. 121101, 2008.

[30] L. Hong and S. E. Page, “Groups of diverse problem solvers can outperform groups of high-ability problem solvers,” *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 46, pp. 16385–16389, 2004.

[31] M. LiCalzi and O. Surucu, “The power of diversity over large solution spaces,” *Manage. Sci.*, vol. 58, no. 7, pp. 1408–1421, 2012.

[32] R. P. Smith and S. D. Eppinger, “A predictive model of sequential iteration in engineering design,” *Manage. Sci.*, vol. 43, no. 8, pp. 1104–1120, 2001.

[33] M. Carrascosa, S. D. Eppinger, and D. E. Whitney, “Using the design structure matrix to estimate product development time,” presented at the ASME Design Engineering Technical Conf., Atlanta, GA, USA, Sep. 13–16, 1998.

[34] A. A. Yassine, D. E. Whitney, J. Lavine, and T. Zambito, “Do-it-right-first-time (DRFT) approach to design structure matrix (DSM) restructuring,” presented at the ASME Int. Design Engineering Technical Conf., Baltimore, MD, USA, Sep. 10–13, 2000.

[35] Q. Yang, T. Yao, T. Lu, and B. Zhang, “An overlapping-based design structure matrix for measuring interaction strength and clustering analysis in product development project,” *IEEE Trans. Eng. Manage.*, vol. 61, no. 1, pp. 159–170, Feb. 2014.

[36] H. C. M. Leon, J. A. Farris, and G. Letens, “Improving product development performance through iteration front-loading,” *IEEE Trans. Eng. Manage.*, vol. 60, no. 3, pp. 552–565, Aug. 2013.

[37] D. D. Campos Silva, L. P. Santiago, and P. M. S. Silva, “Impact of premature information transfer on cost and development time of projects,” *IEEE Trans. Eng. Manage.*, vol. 59, no. 4, pp. 692–704, Nov. 2012.

[38] U. N. Bhat, *Elements of Applied Stochastic Processes*, 2nd ed, Hoboken, NJ, USA: Wiley, 1984.

[39] H. M. Hasanien, “Particle swarm design optimization of transverse flux linear motor for weight reduction and improvement of thrust force,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 4048–4056, Sep. 2011.

[40] V. Krishnan and K. T. Ulrich, “Product development decisions: A review of the literature,” *Manage. Sci.*, vol. 47, no. 1, pp. 1–21, 2001.

Jeffrey W. Herrmann received the B.S. degree in applied mathematics from the Georgia Institute of Technology, Atlanta, GA, USA, in 1990, and the Ph.D. degree in industrial and systems engineering from the University of Florida, Gainesville, FL, USA, in 1993.

He is currently a Professor with the University of Maryland, College Park, MD, USA, where he holds a joint appointment with the Department of Mechanical Engineering and the Institute for Systems Research. He is the Academic Director at the University of Maryland Quality Enhancement Systems and Teams Honors Program.