Biased Information Passing Between Subsystems Over Time in Complex System Design

Citation: Austin-Breneman, Jesse, Bo Yang Yu, and Maria C. Yang. “Biased Information Passing Between Subsystems Over Time in Complex System Design.” Journal of Mechanical Design 138, no. 1 (November 4, 2015): 011101.

As Published: http://dx.doi.org/10.1115/1.4031745

Publisher: ASME International

Persistent URL: http://hdl.handle.net/1721.1/120033

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Biased Information Passing Between Subsystems Over Time in Complex System Design

Jesse Austin-Breneman
Department of Mechanical Engineering,
University of Michigan,
Ann Arbor, MI 48109
e-mail: jausbren@umich.edu

Bo Yang Yu
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA 02139
e-mail: byyu@mit.edu

Maria C. Yang
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA 02139
e-mail: mcyang@mit.edu

During the early stage design of large-scale engineering systems, design teams are challenged to balance a complex set of considerations. The established structured approaches for optimizing complex system designs offer strategies for achieving optimal solutions, but in practice suboptimal system-level results are often reached due to factors such as satisficing, ill-defined problems, or other project constraints. Twelve subsystem and system-level practitioners at a large aerospace organization were interviewed to understand the ways in which they integrate subsystems in their own work. Responses showed subsystem team members often presented conservative, worst-case scenarios to other subsystems when negotiating a tradeoff as a way of hedging against their own future needs. This practice of biased information passing, referred to informally by the practitioners as adding "margins," is modeled in this paper with a series of optimization simulations. Three "bias" conditions were tested: no bias, a constant bias, and a bias which decreases with time. Results from the simulations show that biased information passing negatively affects both the number of iterations needed and the Pareto optimality of system-level solutions. Results are also compared to the interview responses and highlight several themes with respect to complex system design practice. [DOI: 10.1115/1.4031745]

1 Introduction

Large-scale engineering systems require design teams to balance complex considerations using a wide range of design and decision-making skills. Formal approaches for optimizing complex systems offer strategies for arriving at optimal solutions in situations where system integration and design optimization are well-formulated. However, in practice suboptimal results are often reached at the system level. This can be due to many factors: satisficing decision-making [1], time or budget constraints, ill-defined problems [2], or difficulties associated with human-to-human interrelations [3].

Simulation tools can be used to explore the impact of the factors mentioned above. Simpson et al. presented a wide range of problems that can be addressed through these mathematical models and associated algorithms [4]. Simulations are also used to evaluate formal design approaches. Sobieszczanski-Sobieski and Haftka's survey [5] demonstrates the range of applications in the aerospace industry. Common components studied by these simulations are: (1) the team structure or roles, (2) the form of the information passed between subsystems, and (3) how each subsystem makes decisions and tradeoffs.

This paper presents results from a dozen field interviews of subsystem and system-level practitioners within one aerospace organization. The interviews focused on how real-world human decision-making process differed from formal design strategies. The intent was to understand how subsystem designers reach agreement with each other as part of an overall system design, and what strategies designers use in deciding how to share and pass information.

This study consists of two distinct phases. The first uses an interview-based methodology to develop insight and describe the behavior of interdisciplinary design teams performing complex system design in the aerospace industry. Based on the results of the interviews, the second part utilizes formal multidisciplinary optimization (MDO) techniques to simulate the described behavior of subsystems negotiating to a system-level optimum.

This study seeks to answer the following questions:

(1) What strategies do real-world aerospace designers and engineers use when negotiating design parameters with other subsystems?
(2) What impact might these strategies have on system-level optimality?
(3) What impact might these strategies have on the speed of system optimization?

Speed and optimality are important criteria for comparing optimization algorithms and can lead to a better understanding of the impact of the real-world strategies described. The research questions aim to answer the broader questions of whether these strategies are an issue that should be considered and if so can we develop processes robust to this type of behavior?

2 Related Work

This paper draws on previous work in both formal mathematical models of the design process as well as more qualitative studies of team behavior. Perspectives from both are used to gain insight into the effect of biased information passing.

2.1 Complex System Design Process Models. A rich body of literature exists investigating the modeling of the complex system design process. Game theory is one approach for modeling the multidisciplinary design process and was first proposed by Vincent [6] and further developed by Lewis [7] and Whitfield et al. [8]. These traditional game theoretic approaches have also been combined with decision-based design [9] and adopted in a broad range of design research [10–13] to become a prominent framework for the study of multidisciplinary design problems [14]. Game theoretic attempts to identify a rational design (Nash equilibrium [15]) given limits to the amount and form of information being passed between designers. The complex system design process can also be viewed as a multi-objective optimization problem. MDO is one approach which utilizes this philosophy [16]. MDO models generally rely on a system facilitator to make

Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received February 27, 2015; final manuscript received September 28, 2015; published online November 4, 2015. Assoc. Editor: Kristina Shea.

optimal tradeoffs that will benefit the overall system. Design researchers draw from this literature to appropriately model their particular instance of complex system design.

Design research has also considered uncertainty and its propagation through complex systems. Takamatsu used the concept of formal design margins to manage risk throughout the complex system design process [17]. Margins are often defined as probabilistic estimates of the uncertainty of design parameters relative to either worst-case estimates or performance goals. Formal design margins are one replacement for heuristic margins and intuition previously used by design teams. Thunnissen proposed methods for determining these margins and using them to manage risk tolerances [18]. Other researchers have demonstrated the range of applications of these concepts in supporting complex system design [19–21].

2.2 Key Components of Formal Models. Simulations based on these formal models have allowed researchers to observe the effect of changes, at an abstract level, in team structure, information passed, and individual decision-making on performance metrics, such as the speed and accuracy of the optimization. Yi et al. [22], Honda et al. [23], and Martins and Lambe [16] compared different team structures in both game theoretic and MDO approaches. The studied MDO team structures vary from hierarchical formulations where a system integrator makes system-level decisions to nonhierarchical formulations where information flows equally between all actors within the system. Information passing has been studied from both a robustness perspective [24] and the effect of the amount of information on system performance [25]. Collopy outlines a strategy for reaching an optimal design based on passing of gradient information [26]. Lewis and Mistree presented a game theoretic approach, where each agent is involved in the decision-making part of the optimizing task. Agents made decisions using a compromise decision support problem [27]. Robust design also explores the use of uncertainty models in the decision-making process [28]. Limits to the decision-making process have also been described by researchers investigating bounded rationality [29]. In doing this type of analysis, researchers have suggested best practices for design processes.

2.3 Negotiation in Complex System Design. Negotiation in the context of engineering design is a topic with contributions from a variety of fields including design research, management science, economics, and psychology. Smith and Eppinger [30] presented a method utilizing a work transformation matrix to help design teams identify controlling features of a physical design and subsystems that will require more iterations than others. Yassine and Braha [31] presented a method using an information exchange model to help subsystems represent complex task relationships better when negotiating. Yassine et al. [32] examined the phenomena of information hiding in complex system design. This occurs when local subsystem optimization and system-level optimization occur asynchronously and information gained from the local development is hidden from the system-level process. Klein et al. [33] modeled the effect of the team or network structure on the negotiations during the complex system design process. Di Marco et al. [34] examined the effect of individual team member culture on the negotiation process in complex system design teams. This paper draws on these sources to help model the negotiation between subsystems.

2.4 Team Communication. Literature from organizational behavior, psychology, engineering, and sociology have all examined how communication affects team performance [35]. Nardi and Whittaker [36] demonstrated that social communication within a team requires a shared team understanding. Face-to-face communication during distributed design was shown to be particularly important. In collocated teams, design quality was found to be highly dependent on networking in the physical space [37]. Team cognition is a related area in which communication has also been addressed. Cooke and Gorman [38] used team communications to measure the team decision-making process and ability to accomplish high-level processing of information and reach an optimal decision. These lessons have also been used to develop design tools which support teams in communicating more effectively and in different mediums [39]. This paper draws on these works to provide a framework for understanding and modeling team communication in a more effective manner.

2.5 Problem Selection. A key issue in validating and understanding results of simulations of the design process is the selection of test problems. Coello Coello et al. [40] categorized the types of multi-objective optimization test problems and provided an overview of existing test suites. This work is part of a larger body of literature addressing many of the issues involved in developing appropriate test suites [41]. It should be noted that test suites can be useful for comparing and evaluating optimization algorithms but may not be representative of algorithm performance on "real-world" problems. In order to gain the maximum insight from the simulations, a test suite should be comprised of a variety of types of problems. This paper draws from several sources to incorporate as many different types of test problems as possible.

2.6 Research Gap. This paper focuses on the interactions between subsystems in complex system design. Current literature either focuses on improving mathematical formulations of formal models of the design process or developing qualitative frameworks of team behavior. This paper seeks to bridge the gap between the two and use the power of both approaches to gain a better understanding of how subsystems interact in complex system design tasks. In particular, this study hopes to both improve the effectiveness of the simulations by more realistically modeling the social component of human behavior and to improve the qualitative frameworks by quantifying the estimated effect of the human factors.

3 Phase 1: Interviews With Practitioners

3.1 Interview Methods. The interview phase consisted of 12 interviews with lead subsystem designers and system integrators within a large aerospace organization. Subsystem designers were drawn from a diverse set of subsystems, such as structures, propulsion, avionics, guidance and navigation control, materials and manufacturing, systems integration, operations, liquid engines, and testing.

Each interview consisted of an hour of open-ended discussion on system integration management and intersubsystem communication. The primary question asked was, "How do you manage the integration of your subsystem with other subsystems?" Biographical informations, such as job title and description, were also recorded. The interviews were not recorded due to confidentiality. Notes were taken separately by two investigators. Select quotes and themes from the interviews are presented below. These concepts were incorporated into and informed the second phase of the study.

3.2 Interview Results

3.2.1 Finding #1: Structure of Negotiations. The interviewees described a number of modes of interacting with other subsystems. The notable finding is that their patterns of interactions could be characterized fairly well in the formal terminology of MDO and game theoretic models depending on the level of agreement between the subsystems. The basic mode of negotiation followed a game theoretic model, with subsystem designers connecting with their counterparts in other subsystems to manage tradeoffs on an informal level. Larger disputes were negotiated following a hierarchal MDO model with disagreements between subsystems settled by a committee of upper management.

All ten subsystem designers and the two system integrators mentioned direct personal relationships as a conduit for negotiation with designers in the other subsystems they interfaced with regularly. One example of this type of negotiation is the "volume envelope" mechanism. One subsystem set "envelopes" or volumes other subsystems could use as a volume constraint early in the design process. If another subsystem needed more space, the subsystem designer first went to subsystem designers of nearby envelopes to reach a compromise on the volume needed.

A similar negotiation was described with respect to power requirements. Power requirements for one subsystem were negotiated between the appropriate level of subsystem designer early on and then adjustments and compromises were made throughout the process. This was facilitated by the placement of personnel physically near each other. Engineers from other subsystems have offices or "sit" in the relevant subsystem office suite.

Compromises are also facilitated by engineers designated as leads for integrating subsystems. These engineers are representatives from the different subsystems and negotiate at a more formal level during planned meetings. A three-level structure of negotiation was proposed by several of the designers. The lowest level is within the subsystem; this happens routinely on a daily basis and focuses on optimizing the subsystem and setting requirements. Most of the negotiation of tolerances and requirements happens at a cross-cutting second level. Two engineers independently estimated that 80–90% of issues raised were resolved at this level. The third level involves upper management and a formal conflict resolution process. For example, a disagreement between two subsystems which could not be resolved at either of the two lower levels could be brought before the weekly chief engineers meeting and a panel of upper management would then make a decision. These levels were described by multiple participants as "down and in" and "up and out" exemplifying the correlation between level of formality and interaction within or without the team.

The higher level of parameter conflict resolution follows a hierarchical model of negotiation used in some formulations of MDO. Subsystems no longer negotiate between themselves, but bring it to a system integrator who makes a decision. This view was supported in the interviews with the system integrators. One system integrator described his role as "finding problems and fixing them." Another difference between the self-reporting on the levels was the formality. The levels increase in formality with the third level requiring documentation of the conflict and a presentation of both sides of the issue before a panel of upper managers. All such third level conflicts are tracked throughout the process and system integrators are required to resolve them at different major milestones. This is in stark contrast with the informality of the second level at which subsystem designers simply make changes by talking to another subsystem designer. Estimates for the relative amount of problems which reached the third level ranged from 2% to 5%. All subsystem designers expressed their trust in the upper management board to resolve conflicts in an optimal way.

3.2.2 Finding #2: Biased Information Passing Over Time. An important aspect of negotiation that arose in the interviews was the concept of biased information passing. This negotiation tactic was used primarily at the cross-cutting second level between subsystem designers from different subsystems as described above. Interviewees reported using the tactic when deciding the value of a single parameter during repeated negotiations over time. Although this behavior was described by subjects for different types of variables, it does not include negotiating tradeoffs between multiple design variables.

The term "margins" was used by interviewees to refer to this practice of reporting "conservative" parameters to other subsystems during the negotiation process. The subjects' definition of margins is distinct from the formal definition of risk or performance margins detailed in the related work section. In these cases, the "conservative" estimates of the parameters are used as a negotiation tool between subsystems and do not reflect the level of uncertainty attached to the design parameter. The phrase "keeping something in my back pocket" was used independently by a majority of the subsystem designers to describe this issue. For example, one subsystem designer highlighted the use of conservative estimates in the development of the budget for a previous project. The subsystem built an extra 30% cushion into their budget estimate as insurance against future budget cuts. The cushion consisted of "budget off-ramps" or extra tests and tasks that were not strictly necessary and could be cut easily near the end of the project. This structure was due to the subsystem designers belief that they would be later asked to cut down their budget, thus the higher budget at the outset offsetting future losses. One interviewee reported that conservative estimates were one factor which contributed to cost overruns and negative consequences for the project. A similar practice was used with parameters that interfaced between subsystems, such as mass, volume, and estimated time to completion of a task. One of the engineers reported that estimated mass was reported with a 30% cushion at the outset, which was reduced over time to 10% near the final design review to allow for negotiation, thus becoming a decreasing margin over time.

It should be noted that this practice is not necessarily suboptimal and can lead to highly robust systems. However, many of the participants felt that the practice had some negative effects. The most common example raised was both parties being conservative in a negotiation and reaching a highly suboptimal compromise. Some subsystem designers believed large design decisions, such as the switch in the overall structure of one project to a substantially different architecture, were based on overly conservative estimates and led to major cost overruns. System integrators also discussed the difficulty in obtaining accurate information from subsystems. One system integrator discussed how conservative estimates in both the inputs as well as the system models used by the subsystems led to cost and schedule failures. They also reported the use of formal risk mitigation procedures which can be inaccurate when presented with conservative inputs.

4 Phase 2: Simulations of Real-World Behavior

4.1 Simulation Structure. The simulation phase consisted of the development of a series of MDO simulations aimed at recreating and quantifying the themes introduced in the interview process. The main purpose of the simulation phase was to simulate the behavior of biased information passing and quantify the effect on system optimization. Simulations were performed on a two-player system because this represented the scenario in the nonhierarchical second level in which biased information passing was reported to occur and to simplify initial calculations.

The interview results suggested that the organization's design team uses a sequential design optimization architecture, also known as fixed-point iteration [42]. In this portion of the study, a series of optimization simulations were created to mimic this design process. Interviewees reported that the vast majority of resolved conflicts occurred during informal negotiations between two subsystem designers and not in the formal hierarchical process established within the organization. Thus, only a two-player system was considered for demonstration of the core concept. The two-player system consisted of two subsystems (subsystem 1 and subsystem 2) each with their own objective function. Optimization was performed sequentially with subsystem 1 optimizing its design parameters and then passing point design information to subsystem 2. Subsystem 2 then minimized its design parameters based on this information, and subsystem 2 then passed point design information back to subsystem 1 completing a single system iteration. This is presented in Fig. 1.

The concept of biases is introduced in the passing of point design information between subsystems. The proposed concept of bias is distinct from the traditional use of margins from complex system design literature. The new concept of biased information passing proposed by the authors is described in the interviews as occurring in addition to the traditional margins estimated by the subsystem designers based on uncertainty. In this model, the

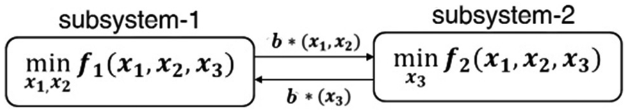
subsystem-1
$\min_{x_1,x_2} f_1(x_1,x_2,x_3)$
$b*(x_1,x_2)$
$b*(x_3)$
subsystem-2
$\min_{x_3} f_2(x_1,x_2,x_3)$

Fig. 1 System schematic for one iteration

original point design information before bias includes the subsystem designer's best estimate. This estimate includes all of the probabilistic uncertainty or safety factors in the definition of margins from previous work, such as Thunnisen [18]. The bias proposed here is in addition to these risk tolerances. The additional bias is an excess capacity used over time as a bargaining chip and is not related to physical reality of the system. A key assumption is that the subsystem designers do not have sufficient information about the behavior of the other subsystems to be able to discern when the information is biased. This may be a poor assumption when the subsystems have worked together for a long period of time or are in closely related fields. Additionally, because this negotiation is happening informally between subsystem designers, the information is not passed to the system integrator who would likely have a better understanding of the larger system behavior.

The simulations were performed in three different conditions: no bias, static bias, and decreasing bias. In the first condition, no bias was used and point design information was passed normally as in traditional MDO processes. In the static bias condition, the point design information was multiplied by 1.3 during the transfer to the other subsystem to reflect an added bias of 30%. This number was chosen based on the estimates reported in the interviews. Each subsystem was in effect biasing the information passed by 30% in the same direction at every iteration. In the decreasing bias condition, the bias was decreased after each system iteration. The design point information was multiplied by $b = 1.3 - .1i$ for $i = 0, 1, 2, 3\ldots$ and $b \geq 1$. This again reflects information reported during the interview process. Subsystem designers reported that the bias was decreased from 30% to 0% in 10% increments at each design review. There are many other negotiation strategies that could be used to determine this additional bias. A proportional method with a time dependence was modeled because it was the strategy described in the interviews. Future work could include investigating other behaviors or functions for this bias.

Additionally, the constant bias and decreasing bias conditions were evaluated in asymmetrically biased systems. In these simulations, only one of the subsystems biased the passed information. For example, an asymmetric bias in subsystem 1 indicates that subsystem 1 multiplied the passed information by the bias factor b and subsystem 2 did not. These test conditions reflected interview results which indicated that different subsystems could have varying levels of bias in their information passing strategies. In particular, less experienced subsystem designers may be less likely to use this negotiation strategy.

All test conditions were simulated on a test suite of 15 two-objective problems drawn from multi-objective evolutionary algorithms by Coello Coello et al. [40] and from a test suite proposed by Deb et al. [41]. This test suite was chosen for its variety in the type of problems provided. It is well-understood that test suites do not necessarily reflect real-world behavior. However, when comparing algorithms test suites can be used to provide a base level of comparison. This was important in this study to allow for comparison between the three conditions.

Comparison between the different conditions was made along two metrics, optimality and speed. These are the two common metrics used for comparing algorithms [40]. Optimality was measured using the normalized distance to Pareto frontier, which is the shortest Euclidean distance between the Pareto frontier and the final system design after satisfying the stopping condition, normalized by the Euclidean distance between the Pareto maximum and minimum [1]. The stopping condition was defined as either convergence for both subsystems $f_1(i) = f_1(i-1); f_2(i) = f_2(i-1)$ or reaching a Nash equilibrium $f_1(i) = f_1(i-2)$; $f_2(i) = f_2(i-2)$. The Pareto frontier for these test problems was often given as an analytical solution in the test suite. If not available, the Pareto frontier was estimated using the MATLAB genetic algorithm function GAmultiobj. Speed was measured by the number of iterations until the stopping condition was met. The minimization of each subsystem was performed using the MATLAB optimization function f_min_con with the interior-point algorithm. Therefore, a system whose final system design has a greater Euclidean distance to the Pareto frontier or which uses more iterations to reach a stopping condition is defined as "less optimal."

Several parameters were varied at each condition. Each simulation was tested using 100 random starting points to check for robustness to initial conditions. The mode value of results from the 100 random trials was used for analysis. The order of sequential optimization was also varied for each testing condition. This checked whether starting each system iteration by optimizing the first or second subsystem changed the behavior of the system.

The system optimization behavior was then analyzed to determine the effect of each testing condition on the performance metrics. The behavior was also compared to the specific problem characteristics, such as types of constraints and objective functions. This analysis is presented in Sec. 5.

4.2 Simulation Results. Simulations were performed on a test suite of problems from evolutionary algorithms by Coello Coello et al. [40] as well as from the test suite provided in Deb et al. [41]. Solution paths for multi-objective problem 4 (MOP4) under the three test conditions are presented as they display the behavior exhibited by many of the test problems. MOP4 was chosen as the display case for two reasons: (1) the number of iterations was relatively small and (2) the Pareto frontier and solution space had the same order of magnitude. These characteristics make MOP4 easy to visualize. However, the behavior demonstrated by MOP4 is representative of the system response to biased information passing shown by a large majority of the other test problems. This is demonstrated in the overall performance figures.

The normalized distances to Pareto frontier and number of iterations from all of the test problems under the symmetric bias conditions are shown in Figs. 2 and 3, respectively. For the normalized distance measurements, a value of zero would indicate a solution directly on the Pareto frontier and a value of 100% would indicate a solution at the normalizing distance. In Fig. 2, three of the problems have values above 100% of the normalizing factor. Their values are displayed in text boxes to accommodate the spread in chart values. Solution paths from the same starting point for MOP4 under the different conditions are shown in the three figures below. The Pareto frontier on each plot is shown as circles. Figure 4 shows the solution path for the no bias condition. Figure 5 shows the solution path in the static bias condition with $b = 1.3$. The final system design in the static bias case was at a normalized distance of 10%, while the no bias and decreasing bias cases ended on the Pareto frontier. Figure 6 shows the solution path in decreasing bias case.

The normalized distance to Pareto frontier and number of iterations from all the test problems under the asymmetric bias conditions is shown in Figs. 7–10. In these figures, the results for each test suite problem with no bias, asymmetric bias in subsystem 1, asymmetric bias in subsystem 2, and symmetric bias are shown. This is done for the constant bias and decreasing bias conditions. For example, Fig. 7 shows results for the normalized distance to Pareto frontier for each test suite problem under those four conditions for a constant bias strategy.

5 Discussion

Several themes emerge from analysis of the results presented above. First, the interview data clearly demonstrate the use of

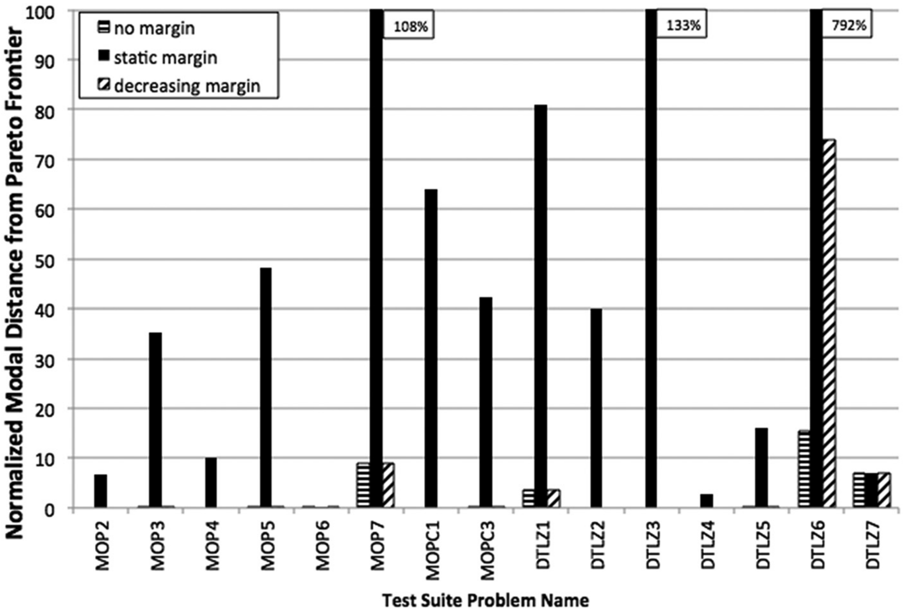

Fig. 2 Normalized distance to the Pareto frontier for all three test conditions

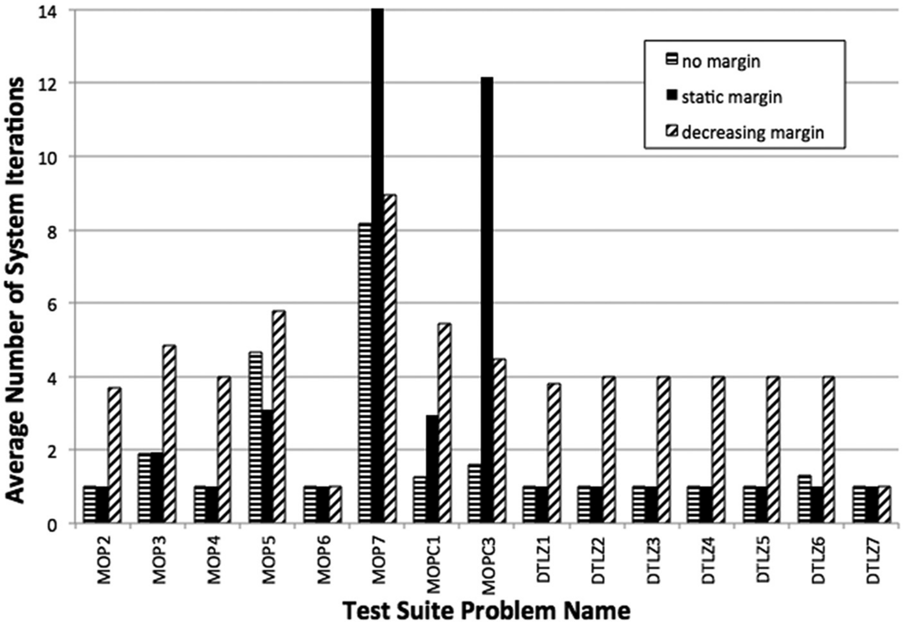

Fig. 3 Average number of system iterations for all three test conditions

biases, and in particular decreasing bias, over time between subsystems in the organization studied. All of the negotiation structures in the organization, both formal and informal, are susceptible to this type of error. However, based on the interview reports, this study focuses on the informal negotiation strategies because they comprise the majority of the resolved negotiations and may also influence outcomes during the formal process. The framework used in the simulations is derived from this information. Second, the use of biases leads to both suboptimal and increased number of iterations in simulations. Third, this behavior was observed across a variety of MOP types and structures.

The use of a decreasing bias strategy was described by almost all of the subsystem engineers and also by the system integrators as a possible cause of system suboptimality. In practice, subsystem engineers report that they provide conservative, worst-case estimates of design parameter, and point design information in discussions with other subsystems. Interviews indicated that this was due to a desire to "underpromise and overdeliver." It may have also be driven by a competition for resources, such as personnel and money between the different subsystems. Decreasing biases is one strategy for ensuring that the subsystem has the resources it needs to complete the required tasks and be robust to unexpected design constraints. Although a risk-averse strategy of using conservative parameters was reported, there exist potential benefits to subsystems which promise performance or delivery times early on in the process even if they cannot meet it. This is especially true in complex system development processes which occur over long periods of time. In these cases, the bias factor could have values which are less than 1. The use of an initial coefficient of $b = 0.3$ and then decreasing over time was motivated by interview results. This is only one possible strategy for biasing information passed. Subsystem designers could use a different

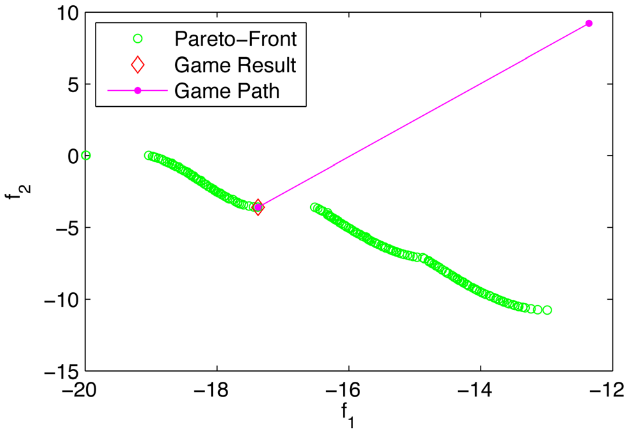

Fig. 4 Solution path in the no bias condition *b* = 0

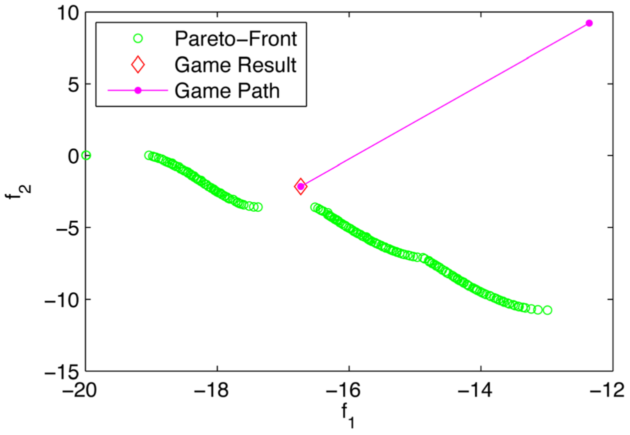

Fig. 5 Solution path in the static bias condition *b* = 1.3

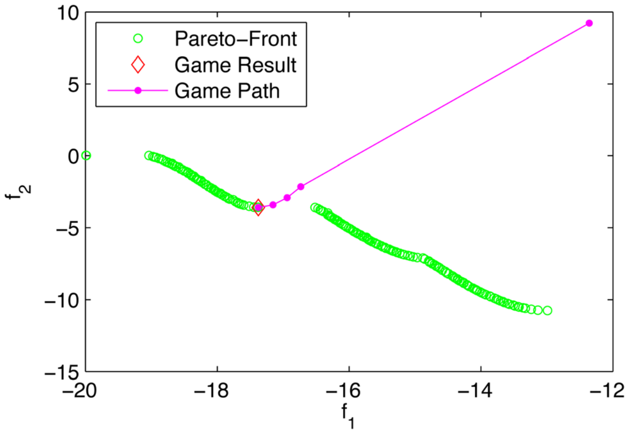

Fig. 6 Solution path in the decreasing bias condition b=1.3−.1 ∗ i

deterministic function for determining bias based on iteration number such as starting at a different initial coefficient or using a nonlinear decrease in bias. Designers could also choose to use a probabilistic method in which the bias coefficient was chosen with some probability of increasing or decreasing.

This can be an effective strategy at the subsystem level, but the simulations demonstrated that it may lead to system-level issues. For example, the system response to different bias conditions in MOP4 shows suboptimal results for the static and decreasing bias conditions. Fig. 4 shows the final system design for the no bias condition to be directly on the Pareto frontier. In Fig. 5, the final system design found using the static bias strategy from the same starting point is further away from the Pareto frontier and clearly less optimal. The decreasing bias condition shown in Fig. 6 did not lead to suboptimal results but did take more iterations. Although commonly used to compare optimization algorithms, the number of iterations is also an important metric when considering the design process. An increased number of iterations reflects a longer overall design process and time is an important resource in any design project. For example, time constraints can be viewed as constraining a design team to a fixed number of design iterations. A team using the decreasing bias strategy may reach a less optimal result given the same number of iterations when compared to a team using no biases, especially if the number of iterations required to reach the Pareto frontier is large. However, given an infinite amount of time and other resources, results from the test suite of problems suggest that the decreasing bias strategy actually may be preferable to the no bias case because it reaches the same level of optimality and the "refinement" period near the end gives the design team more confidence when they explore the area near the Pareto frontier extensively and that feasibility of the results is not sensitive to small changes in inputs.

The system response to the test conditions demonstrated in MOP4 was similar across many of the test suite problems tested. Figure 2 shows how in most of the problems the static bias condition was less optimal than the no bias and decreasing bias conditions. In the two problems which do not fit this pattern, MOP6 and DTLZ7, the structure of the problem caused the optimization algorithm to find the edge of the design space in a single iteration. The boundary of the design space was also on the Pareto frontier. Thus, all conditions found this point and the optimality of the final system design of these problems was insensitive to changes in the bias.

The system response demonstrated in MOP4 was also similar to many of the other test problems with respect to the number of iterations needed to reach a stopping condition, as shown in Fig. 3. The number of iterations needed in the decreasing bias case was also higher than in the other two cases for most of the test problems. In problems whose objective functions were conical, such as MOP5, MOP7, MOPC1, and MOPC3, the behavior was more sporadic. Although it is unclear exactly how the conical structure caused the differences in behavior, the optimization algorithms used many iterations to refine the final system design near the Pareto frontier in the overlap of the two conic sections. The relative size of the static bias to the size of the overlap may have produced a stopping condition either before reaching this refinement stage, such as in MOP5, or kept it in the refinement stage longer as in MOP7, MOPC1, and MOPC3.

The results from the tests of the asymmetric bias conditions showed that the system response was for the test suite considered least optimal in the symmetric case. This is to be expected, since having both people bias the information passed between them would suggest a less optimal scenario. However, it is interesting to observe that many of the test suite problems were sensitive to biased information passing from a particular subsystem. For example, in DTLZ6, the no bias case converged to a solution of a normalized distance of 19% from the Pareto frontier. In the symmetric constant bias case, the normalized distance was 796% from the Pareto frontier. In the asymmetric constant bias condition, subsystem 1 resulted in a normalized distance of 773%, while subsystem 2 resulted in a normalized distance of only 19%. This suggests that almost all of the error in the symmetric case was due to bias in the information passing from subsystem 1. Similar results were found in many of the other test suite problems. This echoes result from the interviews, in which subsystem designers felt that biased information passing from particular subsystems greatly affected the system-level result.

Another notable finding from the asymmetric bias condition results from this test suite is that iterations were not greatly affected. In three of the problems, MOP7, MOPC1, and MOPC3, a particular subsystem caused the optimization algorithm to converge slowly. However, for the majority of cases, the results for asymmetric biasing did not significantly change the number of iterations required for convergence.

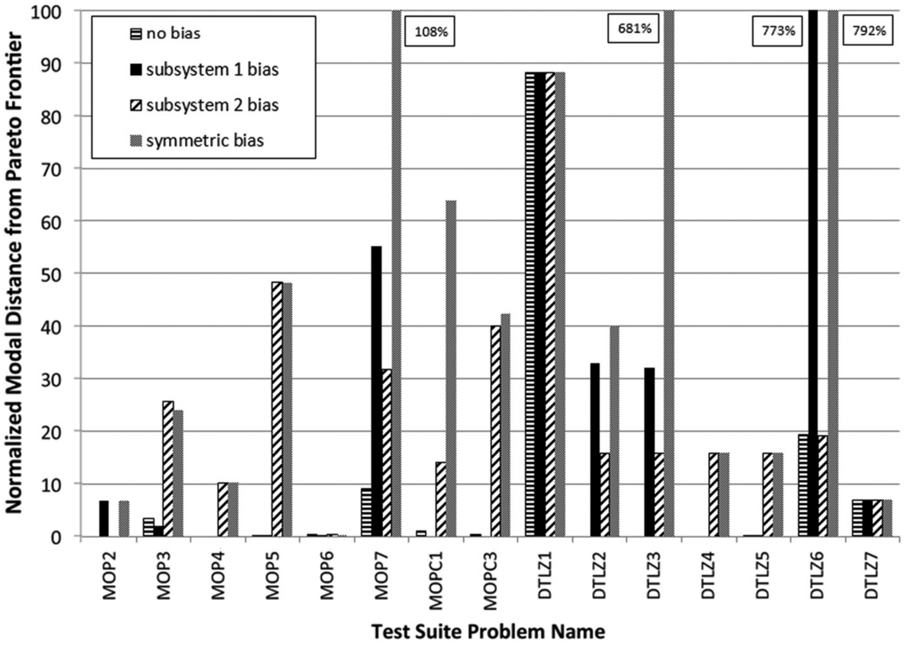

Fig. 7 Normalized distance to Pareto frontier for asymmetric constant bias conditions

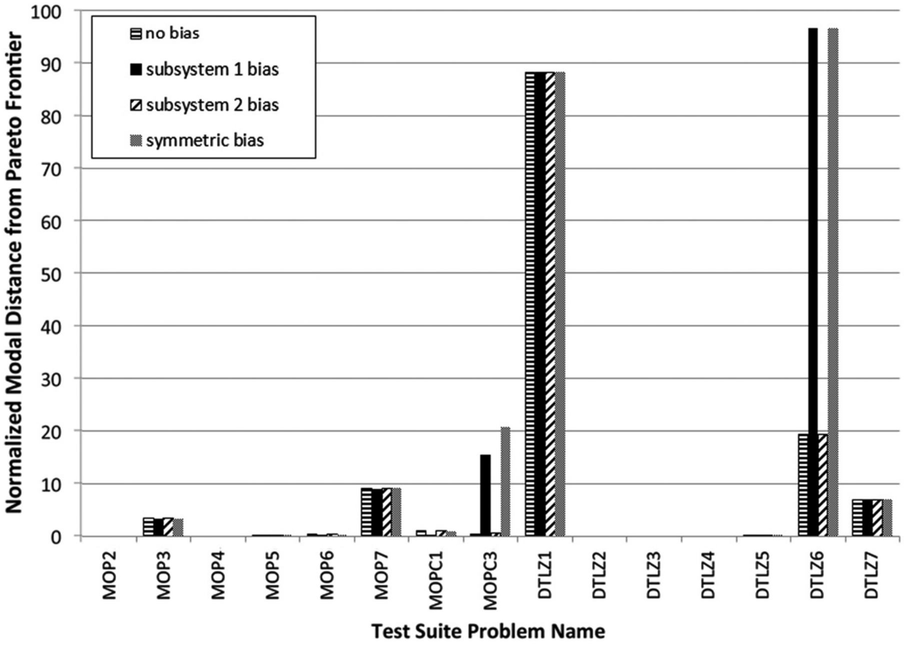

Fig. 8 Normalized distance to Pareto frontier for asymmetric decreasing bias conditions

In practice, subsystem engineers also reported that suboptimal irreversible design decisions were made early in the design process based on biased information from other subsystems. For example, a complicated and expensive structure may be designed and integrated into many subsystems based on mass constraints that are reported early on. The scale of the effect is due to not only the highly connected nature of the subsystems but also the nonlinear nature of the subsystem response to design inputs. Small changes in inputs can have large effects on performance and cost.

This study was limited by several factors. The simulations were performed over a large number of problem types in the two test suites used. However, test suite problems do not necessarily accurately represent algorithm behavior in real-world problems. As such it is difficult to determine what the exact meaning of the increase in the distance from the Pareto frontier or the increase in the number of iterations. However, this simulation does reflect insights provided by the interviewees. This study also only describes behavior reported by members of one organization. The information may not be representative of all design teams working on engineering complex systems.

Finally, this study presents results of a simplified two-player system. Since the two-player case shows that biased information affects the quality of design outcome, it could be argued that biased information passing in a multiplayer system would also have adverse affects on design outcome. However, since the information passing model developed in this study cannot be directly adapted to a multiplayer system, these results may not indicate trends in simulations of larger systems. The suboptimal system-level results reported in the interviews may not be directly or wholly due to biased information passing. The two-player system

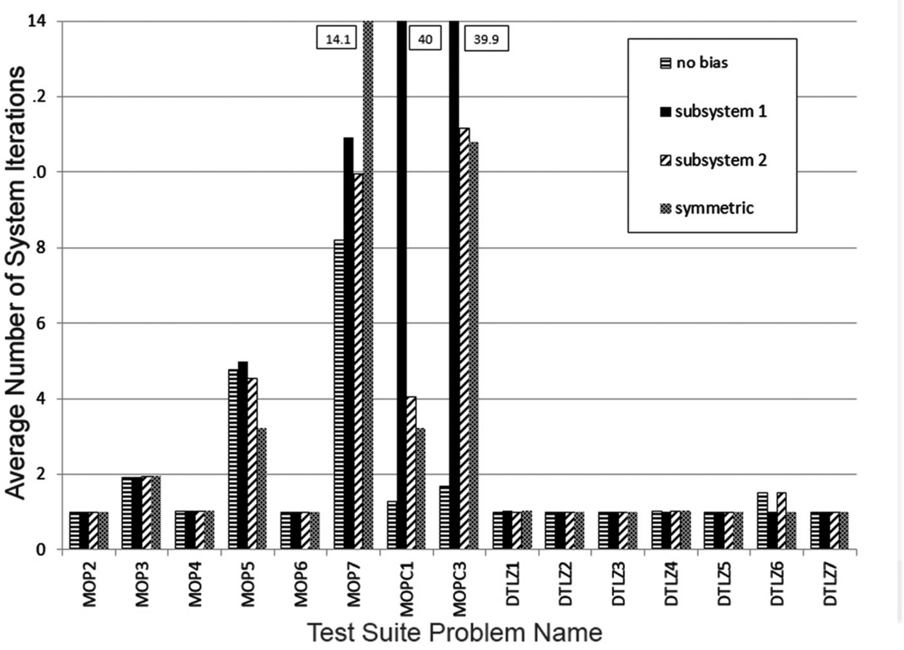

Fig. 9 Number of iterations for asymmetric constant bias conditions

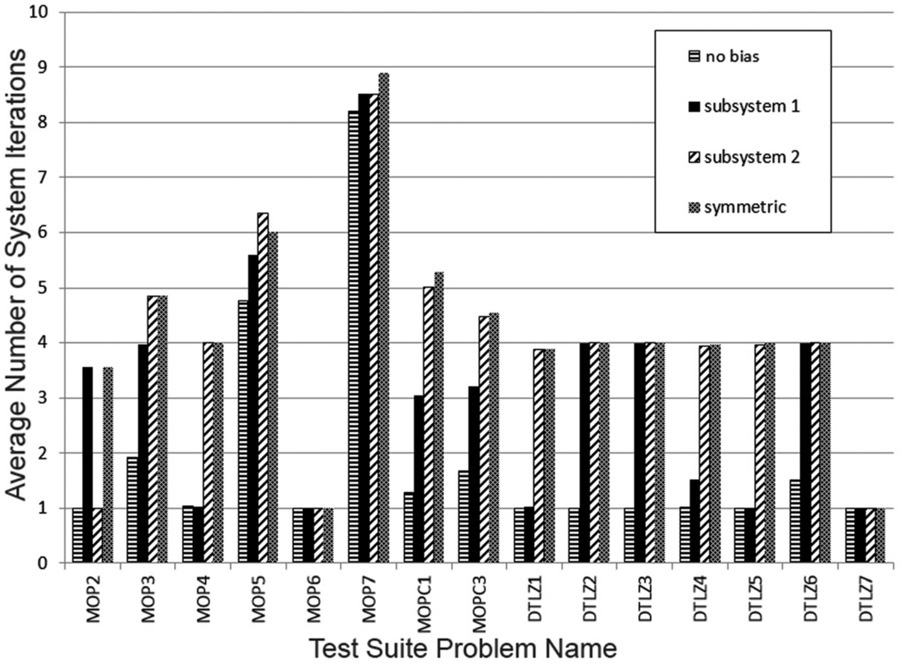

Fig. 10 Number of iterations for asymmetric decreasing bias conditions

model is an initial step in expanding the concept of information biasing to larger systems. For a multi-agent system, a more complex model would need to be developed. The team structure, or how and in what order the subsystems communicate the biased information, would need to be defined. The majority of problems in the test suite used in this study can be easily extended to a multi-agent system. In addition, there may be issues of computational complexity or time with very large multi-agent systems.

6 Conclusions and Future Work

Interview results from a large aerospace organization demonstrated the use of biased information passing at the subsystem level as a negotiation tactic. This behavior reportedly led to suboptimal system-level results. Simulations of three bias conditions showed significant changes in system behavior. Two types of errors were observed regarding speed and optimality

(1) What strategies do real-world aerospace designers and engineers use when negotiating design parameters with other subsystems? Practitioners interviewed reported using both MDO and game theoretic structures for negotiating tradeoffs between subsystems. Lower-level negotiations were done informally in a game theoretic structure, while higher-level negotiations were done formally in front of upper management committees. Interviewees also reported the use of biased information passing between subsystems during negotiations at all levels. The biased information passing was reported as starting at approximately 30% and decreasing over time. Practitioners

also reported asymmetric situations in which the subsystems were biasing the information unevenly.

(2) What impact might these strategies have on system-level optimality?

Although the size of the effect was problem-dependent, biased information passing negatively affected system-level optimality across all problem types tested. Solutions that resulted from strategies incorporating fixed biased information passing negatively affected system-level optimality to a high degree. Solutions resulting from strategies incorporating a decreasing bias had the same level of optimality as those with no bias. Asymmetric bias conditions also negatively affected system-level optimality, but system performance was highly sensitive to which subsystem biased the information.

(3) What impact might these strategies have on the speed of optimization?

The speed as measured by number of system iterations was not affected by the use of a fixed bias in most test problems. However, a decreasing bias strategy increased the number of iterations significantly and the amount increased for more complex problem types. Asymmetric bias conditions had the same affect on the number of iterations as the corresponding symmetric bias condition.

Future work should involve investigating more organizations to see if the use of biased information passing as defined in this study is widespread. Second, the simulations investigating the size of the effect were simplified to a two-player system. The structure within the company for managing negotiations between subsystems seemed to follow a "hybrid MDO–game theoretics" model in which the larger organization follows a hierarchal MDO model, but lower level subsystems adopt a game theoretic approach [43,44]. Braha and Yaneer provided a framework for examining information flow between design tasks and teams in a distributed design environment [45]. Future work should involve simulations of this type to investigate the effect of biased information passing on larger systems. Results found in this study may not reflect the system behavior for the described negotiation strategy across many nodes. Future work should also include real-world problems in which domain knowledge may more heavily impact the decision-making process.

Acknowledgment

We gratefully acknowledge the support of this work by the University of Alabama at Huntsville System Engineering Consortium. This work was also supported in part by a Graduate Research Fellowship from the National Science Foundation.

References

[1] de Weck, O. L., and Jones, M. B., 2006, "Isoperformance: Analysis and Design of Complex Systems With Desired Outcomes," Syst. Eng., **9**(1), pp. 45–61.

[2] Simon, H. A., 1973, "The Structure of Ill Structured Problems," Artif. Intell., **4**(3–4), pp. 181–201.

[3] McGowan, A. M. R., Daly, S., Baker, W., Papalambros, P., and Seifert, C., 2013, "A Socio-Technical Perspective on Interdisciplinary Interactions During the Development of Complex Engineered Systems," Procedia Comput. Sci., **16**, pp. 1142–1151.

[4] Simpson, T. W., Poplinski, J. D., Koch, P. N., and Allen, J. K., 2001, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," Eng. Comput., **17**(2), pp. 129–150.

[5] Sobieszczanski-Sobieski, J., and Haftka, R. T., 1997, "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," Struct. Multidiscip. Optim., **14**(1), pp. 1–23.

[6] Vincent, T. L., 1983, "Game Theory as a Design Tool," J. Mech. Transm. Autom. Des., **105**(2), pp. 165–170.

[7] Lewis, K., 1996, "An Algorithm for Integrated Subsystem Embodiment and System Synthesis," Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA.

[8] Whitfield, R. I., Duffy, R. I., Coates, G., and Hills, W., 2002, "Distributed Design Coordination," Res. Eng. Des., **13**(4), pp. 243–252.

[9] Hazelrigg, G. A., 1998, "A Framework for Decision-Based Engineering Design," ASME J. Mech. Des., **120**(4), pp. 653–658.

[10] Chanron, V., and Lewis, K., 2005, "A Study of Convergence in Decentralized Design Processes," Res. Eng. Des., **16**(3), pp. 133–145.

[11] Chanron, V., Singh, T., and Lewis, K., 2005, "Equilibrium Stability in Decentralized Design Systems," Int. J. Syst. Sci., **36**(10), pp. 651–662.

[12] Xiao, A., Zheng, S., Allen, J. K., Rosen, D. W., and Mistree, F., 2005, "Collaborative Multidisciplinary Decision Making Using Game Theory and Design Capability Indices," Res. Eng. Des., **16**(1–2), pp. 57–72.

[13] Gurnani, A. P., and Lewis, K., 2008, "Using Bounded Rationality to Improve Decentralized Design," AIAA J., **46**(12), pp. 3049–3059.

[14] Lewis, K. E., Chen, W., and Schmidt, L. C., 2006, *Decision Making in Engineering Design*, American Society of Mechanical Engineers, New York.

[15] Nash, J. F., 1951, "Non-Cooperative Games," Ann. Math., **54**(2), pp. 286–295.

[16] Martins, J. R. R., and Lambe, A., 2012, "Multidisciplinary Design Optimization: A Survey of Architectures," AIAA J., **51**(9), pp. 1–53.

[17] Takamatsu, T., Hashimoto, I., and Ohno, H., 1970, "Optimal Design of a Large Complex System From the Viewpoint of Sensitivity Analysis," Ind. Eng. Chem. Process Des. Dev., **9**(3), pp. 368–379.

[18] Thunnissen, D. P., 2004, "Method for Determining Margins in Conceptual Design," J. Spacecr. Rockets, **41**(1), pp. 85–91.

[19] Eckert, C. M., Isaksson, O., and Earl, C. F., 2014, "Design Margins as a Key to Understanding Design Iterations," ASME Paper No. DETC2014-34275.

[20] Sentz, K., and Ferson, S., 2011, "Probabilistic Bounding Analysis in the Quantification of Margins and Uncertainties," Reliab. Eng. Syst. Saf., **96**(9), pp. 1126–1136.

[21] Helton, J., 2011, "Quantification of Margins and Uncertainties: Conceptual and Computational Basis," Reliab. Eng. Syst. Saf., **96**(9), pp. 976–1013.

[22] Yi, S., Shin, J., and Park, G. J., 2008, "Comparison of MDO Methods With Mathematical Examples," Struct. Multidiscip. Optim., **35**(5), pp. 391–402.

[23] Honda, T., Ciucci, F., Lewis, K., and Yang, M., 2010, "A Comparison of Information Passing Strategies in System Level Modeling," ASME Paper No. DETC2010-29026.

[24] Gu, X., Renaud, J., Batill, S. M., Brach, R. M., and Budhiraja, A. S., 2000, "Worst Case Propagated Uncertainty of Multidisciplinary Systems in Robust Design Optimization," Struct. Multidiscip. Optim., **20**(3), pp. 190–213.

[25] Ciucci, F., Honda, T., and Yang, M. C., 2012, "An Information-Passing Strategy for Achieving Pareto Optimality in the Design of Complex Systems," Res. Eng. Des., **23**(1), pp. 71–83.

[26] Collopy, P., 2001, "Economic-Based Distributed Optimal Design," AIAA Paper No. 2001-4675.

[27] Lewis, K., and Mistree, F., 1997, "Modeling Interactions in Interdisciplinary Design: A Game Theoretic Approach," AIAA J., **35**(8), pp. 1387–1392.

[28] Kalsi, M., Hacker, K., and Lewis, K., 2001, "A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex Systems Design," ASME J. Mech. Des., **123**(1), pp. 1–10.

[29] Simon, H. A., 1997, *Models of Bounded Rationality*, MIT Press, Cambridge, MA.

[30] Smith, R. P., and Eppinger, S. D., 1997, "Identifying Controlling Features of Engineering Design Iteration," Manage. Sci., **43**(3), pp. 276–293.

[31] Yassine, A., and Braha, D., 2003, "Complex Concurrent Engineering and the Design Structure Matrix Approach," Concurrent Eng.: Res. Appl., **11**(3), pp. 165–177.

[32] Yassine, A., Joglekar, N., Braha, D., Eppinger, S., and Whitney, D., 2003, "Information Hiding in Product Development: The Design Churn Effect," Res. Eng. Des., **14**(3), pp. 145–161.

[33] Klein, M., Sayama, H., Faratin, P., and Bar-Yam, Y., 2003, "The Dynamics of Collaborative Design: Insights From Complex Systems and Negotiation Research," Concurrent Eng., **11**(3), pp. 201–209.

[34] Di Marco, M. K., Taylor, J. E., and Alin, P., 2010, "Emergence and Role of Cultural Boundary Spanners in Global Engineering Project Networks," J. Manage. Eng., **26**(3), pp. 123–132.

[35] Minneman, S. L., and Leifer, L. J., 1993, "Group Engineering Design Practice: The Social Construction of a Technical Reality," International Conference on Engineering Design (ICED), Vol. 93, pp. 301–310.

[36] Nardi, B., and Whittaker, S., 2002, "The Place of Face-to-face Communication in Distributed Work," *Distributed Work*, P. Hinds, and S. Keisler, eds., MIT Press, Cambridge, MA, pp. 83–109.

[37] Kendon, A., 1990, *Conducting Interaction: Patterns of Behavior in Focused Encounters*, Cambridge University Press, New York.

[38] Cooke, N. J., and Gorman, J. C., 2006, "Assessment of Team Cognition," *International Encyclopedia of Ergonomics and Human Factors*, P. Karwowski, ed., Taylor & Francis Ltd., Boca Raton, FL, pp. 270–275.

[39] Minneman, S., Harrison, S., Janssen, B., Kurtenbach, G., Moran, T., Smith, I., and van Melle, B., 1995, "A Confederation of Tools for Capturing and Accessing Collaborative Activity," Third ACM International Conference on Multimedia, pp. 523–534.

[40] Coello Coello, C. A., Lamont, G. B., and van Veldhuizen, D. A., 2007, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York.

[41] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E., 2005, *Scalable Test Problems for Evolutionary Multiobjective Optimization*, Springer, London.

[42] Brown, N. F., and Olds, J. R., 2006, "Evaluation of Multidisciplinary Optimization Techniques Applied to a Reusable Launch Vehicle," J. Spacecr. Rockets, **43**(6), pp. 1289–1300.

[43] Honda, T., Ciucci, F., Lewis, K., and Yang, M. C., 2010, "A Comparison of Information Passing Strategies in System Level Modeling," ASME Paper No. DETC2010-29026.

[44] Honda, T., Ciucci, F., Lewis, K., and Yang, M. C., 2015, "Comparison of Information Passing Strategies in System-Level Modeling," AIAA J., **53**(5), pp. 1121–1133.

[45] Braha, D., and Yaneer, B., 2007, "The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results," Manage. Sci., **53**(7), pp. 1127–1145.