Proceedings of the ASME 2014 International Design Engineering Technical Conference
& Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, NY, USA

DETC2014-34433

BIASED INFORMATION PASSING BETWEEN SUBSYSTEMS OVER TIME IN COMPLEX SYSTEM DESIGN

Jesse Austin-Breneman *
Bo Yang Yu
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
email: jlab@mit.edu
email: byyu@mit.edu

Maria C. Yang
Department of Mechanical Engineering &
Engineering Systems Division
Massachusetts Institute of Technology
Cambridge, MA 02139
email: mcyang@mit.edu

ABSTRACT

The early stage design of large-scale engineering systems challenges design teams to balance a complex set of considerations. Established structured approaches for optimizing complex system designs offer strategies for achieving optimal solutions, but in practice sub-optimal system-level results are often reached due to factors such as satisficing, ill-defined problems or other project constraints. Twelve sub-system and system-level practitioners at a large aerospace organization were interviewed to understand the ways in which they integrate sub-systems. Responses showed sub-system team members often presented conservative, worst-case scenarios to other sub-systems when negotiating a trade-off as a way of hedging their own future needs. This practice of biased information passing, referred to informally by the practitioners as adding "margins," is modeled 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 to reach 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.

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 sub-optimal results are often reached at the system level. This can be due to many factors: satisficing decision-making [1], time or budget constraints or ill-defined problems [2].

Mathematical simulations are one type of tool used to simulate design space exploration and optimization of complex systems. They can be used to explore the impact of the factors mentioned above. Simpson, et al. present a wide range of problems that can be addressed through these mathematical models and associated algorithms [3]. Simu-

*Corresponding author.

lations are also used to evaluate formal design approaches. Sobieszczanski-Sobieski and Haftka's survey [4] demonstrates the range of applications in the aerospace industry. Key components common to and studied by these simulations are 1) the team structure or roles, 2) the form of the information passed between sub-systems and 3) how each sub-system makes decisions and trade-offs.

This paper presents results from a number of interviews conducted of sub-system and system-level practitioners within one organization in the aerospace industry. The interviews focused on how real-world human decision-making process differed from formal design strategies. The intent was to understand how subsystems would reach agreement with each other as part of an overall system design, and what strategies were used in deciding how to share and pass information.

This study consists of two distinct phases. The first part uses an interview-based methodology to develop insight and describe the behavior of inter-disciplinary design teams performing complex system design in the aerospace industry. Based on the results of the interviews, the second part utilizes formal multi-disciplinary optimization 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 sub-systems?
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 indicators for comparing optimization algorithms and can lead us to a better understanding of the impact of the real-world strategies described. Are these strategies 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 [5] and further developed by Lewis and others [6, 7]. These traditional game theoretic approaches have further been combined with Decision-Based Design [8] and adopted in a broad range of design research [9–12] to become a prominent framework for the study of multidisciplinary design problems [13]. Game Theoretic design attempts to identify a rational design (Nash Equilibrium [14]) 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. Multidisciplinary Optimization (MDO) is one approach which utilizes this philosophy [15]. MDO models often rely on a system facilitator to make optimal trade-offs 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 [16]. 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 [17]. Other reseachers have demonstrated the range of applications of these concepts in supporting complex system design [18, 19].

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. Key components examined in these studies which are common to many of the formal models such as MDO and Game Theoretic are 1) the team structure or roles (i.e. the direction and order in which information is passed), 2) the form of the information passed between subsystems (such as point design and local sensitivities) and 3) how each subsystem makes decisions and trade-offs.

Yi, et al. [20], Honda, et al. [21] and Martins, et al. [15] compared different team structures in both Game Theoretic and MDO approaches. Information passing has been studied from both a robustness perspective [22] and the effect of the amount of information on system performance [23]. Collopy outlines a strategy for reaching an optimal design based on passing of gradient information [24]. 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 [25]. Robust design also explores using uncertainty models in the decision-making process [26]. Limits to the decision-making process have also been described by researchers investigating bounded rationality [27]. In doing this type of analysis, researchers have suggested best practices for design processes.

2.3 Team Behavior

This paper focuses on the second set of components involving information passing. There is a rich body of literature on how communication on a human level affects team performance from organizational behavior, psychology and sociology. Because system design is commonly performed by teams, the most relevant research in this area tests factors which affect team success across an array of interdisciplinary problems. Nardi and Whittaker [28] emphasize the need for a shared team understanding for social communication. They investigated the importance of face-to-face communication in distributed design situations. Similarly, networking in the physical space of collocated teams has been shown to be an important determinant for design quality [29]. Team communication is also addressed in the area of team cognition. Cooke and Gorman [30] demonstrate several measures using communications as a method for understanding the team decision-making process and its ability to accomplish high-level processing of information and reach an optimal decision. This paper draws on these works to provide a framework for understanding and modeling team communication in a more effective manner.

2.4 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 [31] present 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 [32] present a method using an information exchange model to help subsystems represent complex task relationships better when negotiating. Yassine, et al. [33] examines 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. [34] model the effect of the team or network structure on the negotiations during the complex system design process. Di Marco, et al. [35] 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.

2.5 Problem Selection

A key issue in validating and understanding results of simulations of the design process is the selection of test problems. Coello, et al. [36] categorize the types of multi-objective optimization test problems and provide 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 [37]. 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 twelve interviews with lead subsystem designers and system integrators within a large aerospace organization. Subsystem designers were drawn from a diverse set of sub-systems 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 inter-subsystem communication. The primary question asked was, "How do you manage the integration of your sub-system with other sub-systems?" Biographical information 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

Finding #1: Structure of Negotiations

The interviewees describes a number of modes of interacting with other sub-systems. 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 sub-systems. The basic mode of negotiation followed a Game Theoretic model, with sub-system designers connecting with their counterparts in other sub-systems to manage trade-offs on an informal level. Larger disputes were negotiated following an MDO model with disagreements between sub-systems settled by a committee of upper management.

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

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

Compromises are also facilitated by engineers designated as leads for integrating subsystems. These engineers are representatives from the different sub-systems 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 sub-system; this happens routinely on a daily basis and focuses on optimizing the sub-system 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 sub-systems 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 resolution follows a MDO model of negotiation. Sub-systems 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 sub-system designers simply make changes by talking to another sub-system designer. Estimates for the relative amount of problems which reached the third level ranged from 2 to 5%. All sub-system designers expressed their trust in the upper management board to resolve conflicts in an optimal way.

Finding #2: Biased Information Passing Over Time

Another aspect of negotiation that arose in the interviews was the concept of biased information passing. The phrase "margins" was used by interviewees to refer to the practice of reporting "conservative" parameters to other sub-systems during the negotiation process. Their 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 sub-systems and do not necessarily 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 sub-system designers to describe this issue. For example, one sub-system designer highlighted the use of conservative estimates in the development of the budget for a previous project. The sub-system 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 sub-system designers belief 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 sub-systems 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.

It should be noted that this practice is not necessarily sub-optimal, 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 sub-optimal compromise. Some sub-system 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 sub-systems. One system integrator discussed how conservative estimates in both the inputs as well as the system models used by the sub-systems 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 "margins" or biases and quantify the effect on system optimization. Simulations were performed on a two-player system 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 [39]. In this portion of the study a series of optimization simulations were created to mimic this design process. Only a two-player system was considered for demonstration of the core concept. The two-player system consisted of two subsystems each with their own objective function. Optimization was performed sequentially with the first subsystem optimizing its design parameters and then passing point design information to the second subsystem. The second subsystem then minimized its design parameters based on this information. The second subsystem then passed point design information back to the first subsystem completing a single system iteration. This is presented in Figure 1.

The concept of biases is introduced in the passing of point design information between subsystems. The simulations were performed in three different conditions: no

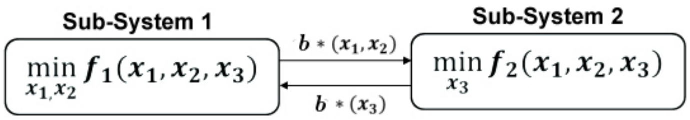

Figure 1. System schematic for one iteration

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 $i = 0,1,2,3\ldots$ & $b \geq 1$. This again reflects information reported during the interview process. Subsystem designers reported the bias was decreased 30% to 0% in 10% increments at each design review.

These three conditions were simulated on a test suite of two-objective problems drawn from Multi-objective Evolutionary Algorithms by Coello, et al. [36] and from a test suite proposed by Deb, et al. [37]. 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 two common metrics used for comparing algorithms [36]. Optimality was measured using the Euclidean distance of the final system design from the Pareto Frontier after satisfying the stopping condition. 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 calculated 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.

Several parameters were varied at each condition. A variety of starting points were tested for each condition and test problem to check for robustness to initial conditions. The order of sequential optimization was also varied for each testing condition. This checked whether having the first or second subsystem optimize first in each system iteration changed the behavior of the system. The system optimization behavior was then analyzed to determine what the effect of each testing condition was 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 the results and discussion sections.

4.2 Simulation Results

Simulations were performed on a test suite of problems from Evolutionary Algorithms by Coello et al. as well as from the test suite provided in Deb, et al. [37]. Solution paths for Multi-Objective Problem 4 (MOP4) under the three test conditions are presented as they display 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.

Results from all of the test problems for Euclidean distance and number of iterations are shown in Figure 2 and Figure 3 respectively. The number of iterations was averaged over 50 random starting points. The Euclidean distance to the Pareto Frontier was normalized by the Euclidean distance between the Pareto maximum and minimum [1]. 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 Figure 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 10% of the normalized distance from the Pareto Frontier, while the no bias and decreasing bias cases ended on the Pareto Frontier. Figure 6 show the solution path in decreasing bias case.

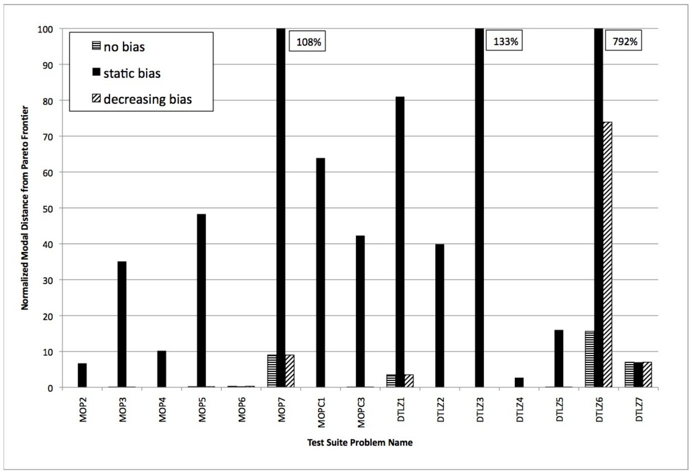

Figure 2. Normalized distance to the Pareto Frontier for all three test conditions

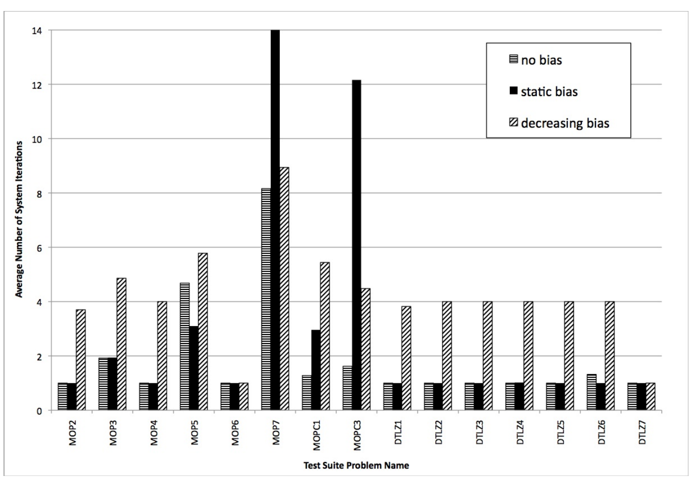

Figure 3. Average number of system iterations for all three test conditions

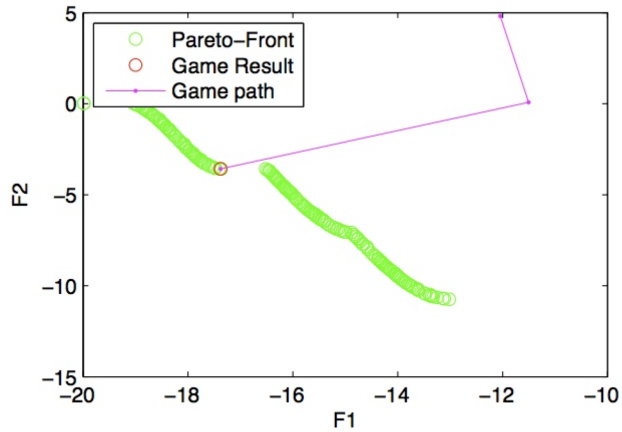

Figure 4. Solution path in the no bias condition. $b = 0$

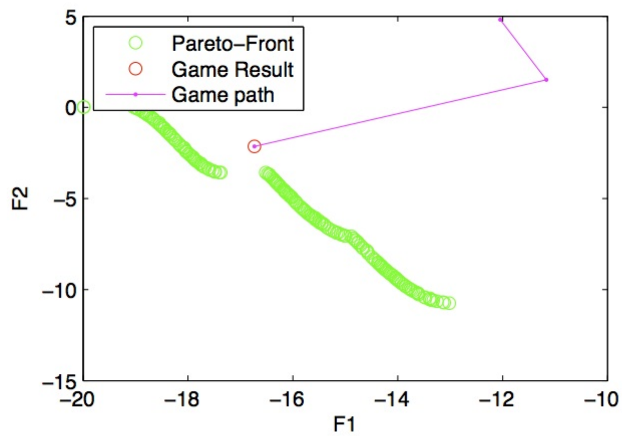

Figure 5. Solution path in the static bias condition. $b = 1.3$

5 DISCUSSION

Several themes emerge from analysis of the results presented above. First, the interview data clearly demonstrates the use of 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. 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 multi-objective problem types and structures.

The use of a decreasing bias strategy was described by almost all of the subsystem engineers and also by

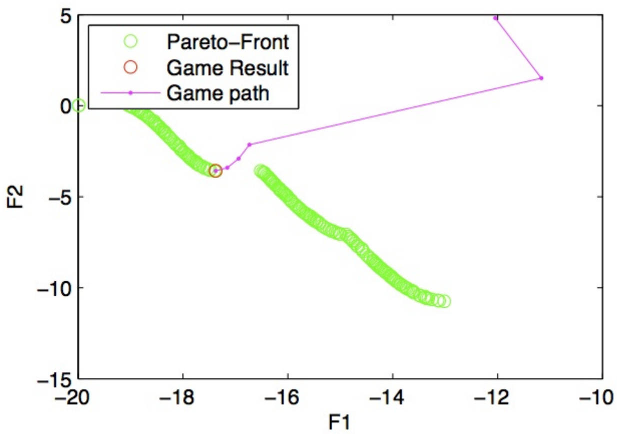

Figure 6. Solution path in the decreasing bias condition. $b = 1.3 - .1 * i$

the system integrators as a possible cause of system sub-optimality. 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 under-promise and over-deliver. 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 the sub-system has the resources it needs to complete the required tasks and be robust to unexpected design constraints.

This can be an effective strategy at the subsystem level, but the simulations demonstrated that it may lead to system-level issues. For example, Figure 4 shows the final system design to be directly on the Pareto Frontier. In Figure 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 Figure 6 did not lead to sub-optimal 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, 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 that they are still in the feasible region.

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. 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. Problems whose objective functions were conic, such as MOP5, MOP7, MOPC1, and MOPC3, the behavior was more sporadic. Although it is unclear exactly how the conic structure caused the differences in behavior, the optimization algorithms used many iterations refining 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 producing 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.

In practice, subsystem engineers also reported that sub-optimal 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 not only the highly-connected nature of the subsystems but also the non-linear 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 multi-player 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 multi-player system, these results may not indicate trends in simulations of larger systems. The sub-optimal system-level results reported in the interviews may not be directly or wholly due to biased information passing. The two-player system 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

Results demonstrated use of biased information passing throughout the organization studied at the subsystem level. This reportedly led to sub-optimal system-level results. Simulations of three conditions: no bias, fixed bias and decreasing bias showed significant changes in system behavior with the addition of biases. 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 sub-systems?

   Practitioners interviewed reported using both MDO and Game Theoretic structures for negotiating trade-offs between sub-systems. 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 sub-systems during negotiations at all levels.

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

   Although the size of the effect was problem-dependent, biased information passing negatively effected 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.

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.

Future work should involve investigating more organizations to see if the use of biased information passing as defined in this study is widespread. Secondly, the simulations investigating the size of the effect were simplified to two-player systems. Future work should involve simulations of larger systems as well as real-world problems.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of this work by the University of Alabama at Huntsville System Engineering Consortium. The work was also supported in part by a Graduate Research Fellowship from the National Science Foundation. The opinions, findings, conclusions and recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] de Weck, O.L. and M.B. Jones. (2006). "Isoperformance: Analysis and Design of Complex Systems with Desired Outcomes." Systems engineering, 9(1), 45-61.

[2] Simon, H.A. (1973). The structure of ill structured problems. Artificial Intelligence, 4(3-4), 181-201.

[3] Simpson, T. W., J. D. Poplinski, et al. (2001). "Metamodels for Computer-based Engineering Design: Survey and recommendations." Engineering with Computers 17(2): 129-150.

[4] Sobieszczanski-Sobieski, J. and R. T. Haftka (1997). "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments." Structural and Multidisciplinary Optimization 14(1): 1-23.

[5] T. L. Vincent (1983). "Game Theory as a Design Tool." Journal of Mechanism, Transmissions, and Automation in Design 105: 165-170.

[6] K. Lewis (1996). An Algorithm for Integrated Subsystem Embodiment and System Synthesis. PhD thesis, Georgia Institute of Technology.

[7] Whitfield,R. I. , Duffy, R. I., et al. (2002). "Distributed design coordination." Research in Engineering Design 13: 243-252.

[8] G. A. Hazelrigg (1998). "A framework for decision-based engineering design." Journal of Mechanical Design 120: 653-658.

[9] Chanron, V. and K. Lewis (2005). "A study of convergence in decentralized design processes." Research in Engineering Design 16(3):133-145.

[10] Chanron, V., Singh, T. and K. Lewis (2005). "Equilibrium stability in decentralized design systems." International Journal of Systems Science 36(10):651-662.

[11] Xiao, A., Zheng, S., and et al. (2005). "Collaborative Multidisciplinary Decision Making using Game Theory and Design Capability Indices." Research in Engineering Design 16(1-2):57-72.

[12] Gurnani, A. P., and K. Lewis (2008) "Using Bounded Rationality to Improve Decentralized Design." AIAA Journal 46(12): 3049-3059.

[13] Lewis, K. E., Chen, W. and L. C. Schmidt (2006). Decision Making in Engineering Design. American Society of Mechanical Engineers.

[14] J. F. Nash (1951). "Non-Cooperative Games. " Annals of Mathematics, 54, pp. 286-295.

[15] Martins, J.R.R, and A. Lambe.(2012). "Multidisciplinary Design Optimization: A Survey of Architectures." AIAA Journal, 1-53.

[16] Takamatsu, T., Hashimoto, I. and H. Ohno.(1970) "Optimal Design of a Large Complex System from the viewpoint of Sensitivity Analysis." Ind. Eng. Chem. Process Des. Develop., 9(3), 368-379.

[17] Thunnissen, D.P. (2004). "Method for Determining Margins in Conceptual Design." Journal of Spacecraft and Rockets, 41(1), 85-91.

[18] Sentz, K. and S. Ferson. (2011). "Probabilistic bounding analysis in the Quantification of Margins and Uncertainties." Reliability Engineering & System Safety, 96(9), 1126-1136.

[19] Helton, J. (2011). "Quantification of margins and uncertainties: Conceptual and computational basis." Reliability Engineering & System Safety, 96(9), 976-1013.

[20] Yi, S., J. Shin, et al. (2008). "Comparison of MDO methods with mathematical examples." Structural and Multidisciplinary Optimization 35(5): 391-402.

[21] Honda, T., F. Ciucci, et al. (2010). "A Comparison of Information Passing Strategies in System Level Modeling." International Design Engineering Technical Conferences. Montreal, Canada.

[22] Gu, X., Renaud, J. et al. (2000)."Worst case propagated uncertainty of multidisciplinary systems in robust design optimization." Structural and Multidisciplinary Optimization, 20(3), 190-213.

[23] Ciucci, F., Honda, T., et al.(2012). "An information-passing strategy for achieving Pareto optimality in the design of complex systems." Research in Engineering Design, 23(1), 71-83.

[24] P. Collopy (2001). "Economic-based Distributed Optimal Design." AIAA SPACE 2001 Conference and Exposition. Albuquerque, NM

[25] Lewis, K. and F. Mistree (1997). "Modeling Interactions in Interdisciplinary Design: A Game Theoretic Approach." AIAA Journal 35(8): 1387-1392.

[26] Kalsi, M., Hacker, K., et al.(2001)."A comprehensive robust design approach for decision trade-offs in complex systems design." Journal of Mechanical Design, 123, 1.

[27] H. A. Simon (1997). Models of Bounded Rationality, MIT Press, Cambridge, MA.

[28] Nardi, B. and S. Whittaker (2002). The Place of Face-to-face Communication in Distributed Work. Dis-

tributed Work. P. Hinds and S. Keisler. Cambridge, MA, MIT Press: 83-109.

[29] A. Kendon (1990). Conducting Interaction: Patterns of Behavior in Focused Encounters. New York, NY, Cambridge University Press.

[30] Cooke, N. J. and J. C. Gorman (2006). Assessment of Team Cognition. International Encyclopedia of Ergonomics and Human Factors. P. Karwowski. UK, Taylor & Francis Ltd.: 270-275.

[31] Smith, R.P., Eppinger, S.D. (1997). "Identifying controlling features of engineering design iteration."' Management Science, 43 (3), pp. 276-293.

[32] Yassine, A. and Braha, D. (2003)." Complex Concurrent Engineering and the Design Structure Matrix Approach."' Concurrent Engineering: Research and Applications. Vol. 11 (3). 165-177.

[33] Yassine A., Joglekar N., Braha, D., Eppinger S., and Whitney, D.(2003)."Information Hiding in Product Development: The Design Churn Effect."' Research in Engineering Design. Vol. 14(3). 131-144.

[34] Klein, M., Sayama, H., Faratin, P., and Bar-Yam, Y. (2003). The dynamics of collaborative design: insights from complex systems and negotiation research. Concurrent Engineering, 11(3), 201-209.

[35] Di Marco, M.K., Taylor, J.E. et al. "Emergence and Role of Cultural Boundary Spanners in Global Engineering Project Networks."'

[36] Coello Coello, C. A., Lamont, G. B., et al. (2007). Evolutionary algorithms for solving multi-objective problems. Springer, New York, NY.

[37] Deb, K., Thiele, L., et al.(2005). Scalable test problems for evolutionary multiobjective optimization. Springer, London, UK.

[38] Devendorf, E., and K. Lewis. (2008). "Planning on Mistakes: An Approach to Incorporate Error Checking into the Design Process," ASME Design Engineering Technical Conferences. New York, NY.

[39] Brown, N.F. and J.R. Olds. (2006). "Evaluation of multidisciplinary optimization techniques applied to a reusable launch vehicle." Journal of Spacecraft and Rockets, 43(6), 1289-1300.

