Designing the Characteristics of Design Teams via Cognitively Inspired Computational Modeling

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Mechanical Engineering

Christopher C. McComb

B.S., Civil Engineering, California State University, Fresno
B.S., Mechanical Engineering, California State University, Fresno
M.S., Mechanical Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

August 2016

Acknowledgements

The four years that I've spent at Carnegie Mellon can be measured in many ways: the lines of code that I've written, the cups of coffee that I've consumed, and the hours that I haven't slept, to name a few. As much as I like to quantify things, all of those metrics fade away as inconsequential when somebody asks me about my experience at CMU. The truth is, the single most important factor in my overwhelmingly positive experience has been the tutelage of two outstanding advisors: Jonathan Cagan and Kenneth Kotovsky. Jon, thank you for knowing when I needed to work through something on my own and when I needed hands-on guidance. Your skill for coaxing the best work out of me showed me the caliber of research that I'm capable of. Ken, thank you for always coming up with the most insightful and unexpected questions. Your comments over the last four years have helped me to recognize my strengths as a researcher. I owe you both a profound debt of gratitude.

I am also extremely thankful for the patience and guidance of my committee members, Kenji Shimada and Kristina Shea, whose feedback has given me access to an external viewpoint that has proven invaluable.

A great deal of thanks is also due to the past and current members of the Integrated Design Innovation Group: Brian Sylcott, Bryony DuPont, Matt Wood, Paul Egan, Noah Tovares, Kosa Goucher-Lambert, Ut Na Sio, Mark Whiting, Shraddha Joshi, Dan Clymer, and Josh Gryory. The quality of my work has been immeasurably improved through your suggestions, critiques, and encouragement. To Bryony, thank you in particular for being a fantastic cubicle-mate at NREL and for convincing me to apply to CMU in the first place. To Kosa, thank you for pacing me every step of the way, and for being an outstanding collaborator in matters both academic and social.

I am grateful to all of the educators and researchers who helped me along my way to Carnegie Mellon, but I would like to single out a few of them with specific praise. To Fariborz Tehrani, thank you taking me under your wing as a naïve undergraduate and for trusting me to make mistakes and break a few things. Your continued mentorship and trust has helped me to develop true confidence in my abilities as a researcher and an engineer. To Mike Lawson and Yi-Hsiang Yu, thank you for patiently guiding me through my first computational research project on wave energy converters and for showing me how computational approaches can enrich research. To Dan Deibert, thank you for showing me just how fascinating the world is, and how much more fascinating it can be when one has an understanding of the science involved. To Gail and David Hutchinson, thank you for pushing me outside of my academic comfort zone and for encouraging my curiosity.

Of course, none of this would have been remotely possible without the support of my family. To my parents Rick and Lee, thank you showing me how to be resourceful, persistent, self-sufficient, and above all things, empathic. I wouldn't be writing this dissertation if it weren't for you. To my sister Marci and her family, thank you for giving me a second home to turn to when I needed to escape from Pittsburgh for a weekend. Most importantly, thank you to my Grandmother for teaching me that no breakfast is complete without pie – a lesson that has just as much value metaphorically as it does literally.

Thank you to my college roommates and long-suffering friends, Christopher "Smalls" Craig and Christopher "Red" Ingle. You have always been there to remind me what life is like outside of grad school, and for that I am exceptionally grateful. To Tim Ishii, your friendship is vital to my continued sanity - thank you for always being so fiercely loyal, and for having the

answer to nearly every one of my problems. Finally, thank you to Lauren Dombrowski for reminding me to breathe, to eat, to love, and to do all those other things so essential to life.

This dissertation is dedicated to every one of you; thank you all for being exactly who you are.

In closing, I also gratefully acknowledge the funding of this research through the National Science Foundation Graduate Research Fellowship, the G. Sundback Graduate Fellowship, and the Air Force Office of Sponsored Research through Grants FA9550-12-1-0374 and FA9550-16-1-0049.

Abstract

Teams are a ubiquitous part of the design process and a great deal of time and effort is devoted to managing them effectively. Although teams have the potential to search effectively for solutions, they are also prone to a number of pitfalls. Thus, a greater understanding of teams is necessary to ensure that they can function optimally across a variety of tasks. Teams are typically studied through controlled laboratory experiments or through longitudinal studies that observe teams *in situ*. However, both of these study types can be costly and time-consuming. Months, if not years, pass between the initial conception of a study and the final analysis of results. This work creates a computational framework that efficiently emulates human design teams, thus facilitating the derivation of a theory linking the properties of design problems to optimized team characteristics, effectively making it possible to design design teams.

This dissertation first introduces and validates the Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework. The central structure of CISAT is modeled after simulated annealing, a global optimization algorithm that has been shown to effectively mimic the problem-solving process of individuals. Specifically, a multi-agent analog of simulated annealing is used in CISAT to mimic the behavior of teams. Several additional components, drawn from the psychology and problem-solving literature, are then included in the framework to enable a more accurate description of individual activity and interaction within the team.

CISAT is then used to investigate the relationship between design problem properties, team characteristics, and task performance. Multiple computational simulations are conducted in which simulated teams with various characteristics solve a variety of different configuration problems. These simulations are then post-processed to produce a set of equations that make it

possible to predict optimal team characteristics based on problem properties, thus enabling the optimal design of design teams. To validate these equations a behavioral study is designed and conducted in which teams of engineering students interact at different frequencies while designing a complex system. Results of the study offer a limited validation of the predictive equations.

This dissertation further highlights the resource efficiency and versatility of CISAT by demonstrating its use in two additional applications. In the first, a new numerical optimization algorithm is derived directly from CISAT by stripping away all but the most quintessential team-based characteristics. The team-based characteristics of this algorithm allow it to achieve high performance across a variety of objective function with diverse topographies. In the second application, CISAT is used in conjunction with Markov concepts to examine the order in which designers make changes to their solutions. Although it has been demonstrated that humans apply changes in a specific order (called a sequence) when solving puzzles, such patterns have not been examined for engineers solving design problems. It is shown that operation sequences are used by designers, and improve solution quality.

This dissertation demonstrates how characteristics of individual designers and design teams can be captured and accurately reproduced within a computational model to advance our knowledge of design methodology. Future extensions of this work have the potential to inform a deeper and more holistic understanding of the search process.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

In the study of problem-solving, as in all science,
there is a right way and a wrong way to go about it
and the correct way may lie somewhere in between.

Kenneth Kotovsky

1.1 Motivation

Changes in workplace culture over recent decades have precipitated a broad shift from individual work to more team-centric workflows [1,2]. The increased use of teams in industry has been specifically noted in both engineering and business [3,4]. Teams are also increasingly common in academic research - this trend is evident in research in science and engineering, the social sciences, and the arts and humanities [5]. Additionally, the relative frequency of single-author patents is dropping [5], demonstrating that intellectual property generation is increasingly a result of teams rather than individuals.

Even with this robust increase in the prevalence of teams, it is difficult to find a single, unified definition for the word *team* itself. Rather, definitions are usually tailored to the application at hand (for example, see definitions in [6–9]). Despite this lack of agreement, two common themes exist at the root of most definitions: multi-agency and communication. *Multi-agency* is the fundamental composition of a team from multiple individuals, and *communication* in this context refers to the ability of those individuals to share information with one another. The balance between these two concepts hints at some of the reasons for the popularity of teams.

For instance, the information processing capacity of individuals is fundamentally limited [10,11], but if multiple individuals communicate and work together as a team they can process more information, allowing teams to tackle larger and more complex problems than individuals alone are capable of. However, the benefit derived from teamwork can exceed the basic multiplication of information-processing abilities. For instance, teams are capable of adapting and responding to dynamic scenarios more quickly than individuals [12]. Teams are also capable of examining the problem at hand from diverse perspectives and simultaneously pursuing different solutions to the same problem. This enables a controlled transition from divergent search and exploration of the design space to convergence on and exploitation of the most fruitful regions. The benefit of this transition from exploration to exploitation is particularly relevant in engineering design, and has analogical implications for design automation.

The above attributes indicate that teams should be more effective than non-interacting individuals, and the superiority of teams is also generally assumed in practice [13]. The superior performance of teams has been demonstrated in domains including concept evaluation and selection [14]. However, there are a number of instances in which teams have been demonstrated to perform less effectively than non-interacting individuals [15,16]. The inferior performance of teams in these cases may be attributed to a number of phenomena. For instance, the process of converging on a common solution may, if over-emphasized, lead to a cognitive state known as groupthink [17]. In this state, the development of consensus within a team is given more importance than the search for quality solutions, which can lead to low team performance [18]. It has also been demonstrated that teams make more risky decisions than individuals [19], an attribute of teams which can be particularly troublesome in safety-critical applications. Further, in a process known as social loafing, individuals may exert less effort when they work as part of a team than when they work alone [20]. Therefore, teams may produce less aggregate effort than the same number of individuals working independently.

Some research has also shown that the best approach for solving a problem may be dependent on the characteristics of that problem [21,22]. This means that a team may not perform at its full potential if it does not select an approach that is appropriate to the problem at hand, or even in some cases that the best approach may not be a traditional interacting team. A greater understanding of the relationship between task performance, team characteristics, and problem properties is necessary to ensure that teams perform optimally across a range of problems.

Teams are traditionally examined through experimental cognitive studies in which teams of humans solve a task. While such studies produce data with high veracity, they can be limited by long cycle times - from conceptualization to data collection and analysis, a standard cognitive study may take anywhere from several months to over a year to conduct. The time required to conduct a study may be compounded by the number of conditions or treatments involved. This potentially limits the scope of research projects. For instance, it could be valuable to study the impact of various properties of design problems on the design process, and potentially to optimize different team approaches for solving problems with specific characteristics. However, the number of treatments necessary in such a cognitive study would be excessive, and the task would become infeasible.

Using traditional cognitive studies may also make it impossible to infer the underlying thought processes that led to specific design decisions or specific solution attributes, and it may be difficult to isolate the effects of specific cognitive phenomena. For example, consider the way in which humans learn sequences of actions. This process can take place either explicitly through focused effort [23,24] or passively without the need for direct attention [25,26]. Because sequence learning can occur passively, it becomes challenging (if not impossible) to perfectly isolate the effects of sequence learning using traditional cognitive studies.

A variety of computational models have been developed to support and supplement traditional cognitive research of human teams. Many of these models are exceedingly complex [27,28], making results difficult to interpret and the models themselves unwieldy to use. Other, simpler models have provided results that align qualitatively with the literature, but those models were not applied to real-world problems and thus cannot readily be validated [29–32]. There is a need for a computational model that provides a test bed for various phenomena while remaining elegantly simple and directly applicable.

Simulated annealing is a well-known optimization algorithm that has been used to model both individual problem-solvers [33] and individual designers [34]. The simulated annealing algorithm is based on the process of physically annealing materials to remove residual stresses [35]. In the physical annealing process, atoms initially move randomly, but as the material is cooled their movements become more deterministic until they reach a minimum energy configuration. In the analogous computational process, changes to a design solution are initially made with a high degree of randomness, but the algorithm progressively transitions to more deterministic search patterns until a solution is achieved. This progressive transition from stochastic to deterministic search is echoed in the activities of human problem-solvers, enabling simulated annealing to effectively model patterns of human problem-solving [33]. Simulated annealing will be used in this dissertation as one key feature in the creation of a computational model of human teams.

1.2 Thesis Statement

The increasing prevalence of teams across many domains necessitates greater research into team-based activities, specifically regarding how teams can be guided towards optimal or improved performance. However, common methods for researching teams are slow and expensive. Computational models designed to supplement traditional studies are either excessively complex or lack experimental validation. The following thesis is presented as a possible remedy for these issues:

> Capturing characteristics of human teams within an accurate computational framework enables the derivation of a theory linking problem properties to optimized team characteristics, at a scope that would prove infeasible with empirical studies alone.

The defense of this thesis begins with a cognitive study that demonstrates additional motivation for the work. This is followed by the creation of a computational model of engineering design teams. The model is developed with the explicit goals of enabling direct applicability to engineering design problems while maintaining relative model simplicity. It is shown that this model is resource-efficient and provides an accurate representation of design teams. A large array of computational simulations is then conducted in order to derive a theory that relates the properties of design problems to the best human team characteristics to use for solving those problems. In essence, this theory provides an approach for the optimal design of design teams. A prediction made using this theory is then validated in a cognitive study. Two additional chapters share research that was enabled by the team-based modeling framework developed here, highlighting its versatility.

1.3 Organization

The organization of this dissertation is shown graphically in Figure 1.1. The thesis stated above is specifically supported in Chapters 2 through 5. In Chapters 6 and 7, additional research is presented that was enabled by the computational model of design teams developed in Chapter 3.

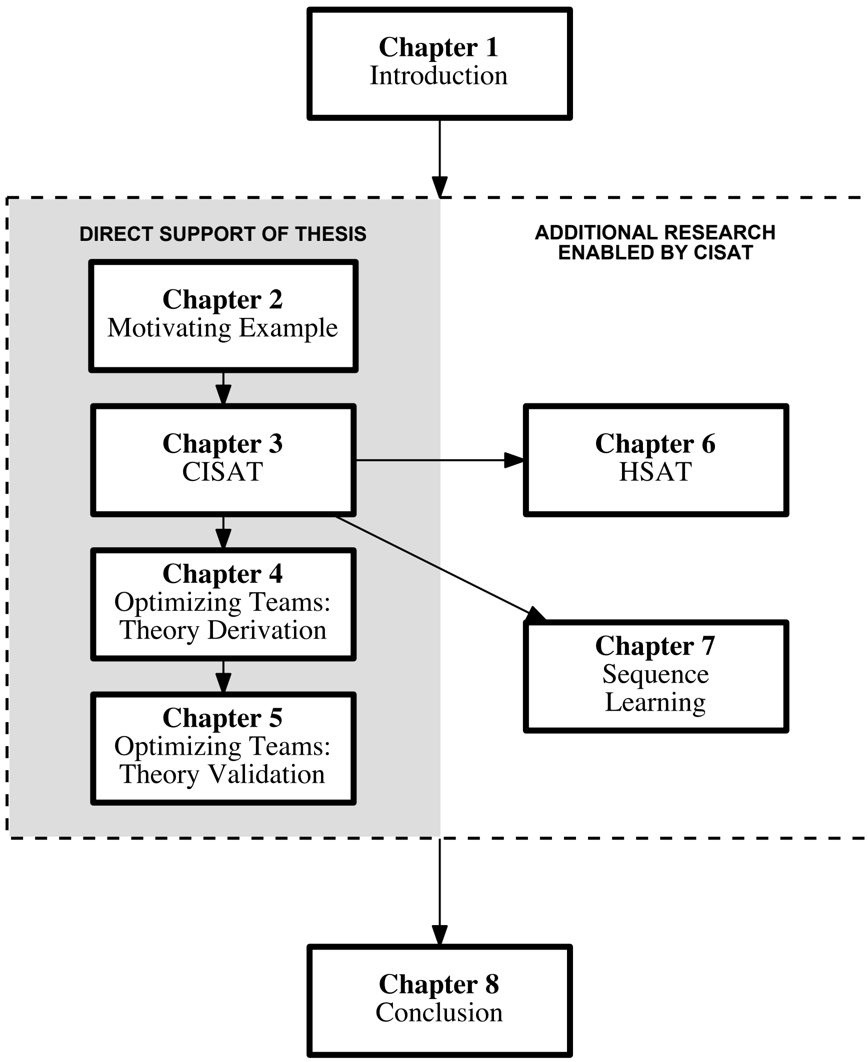


FIGURE 1.1. ORGANIZATION OF THE CHAPTERS IN THIS DISSERTATION.

The results of a cognitive study are introduced in **Chapter 2** and used to further motivate the work contained in this dissertation. At the core of this work is the creation of the Cognitively-Inspired Simulated Annealing Team (CISAT) computational framework, which is detailed in **Chapter 3**. CISAT uses simulated annealing constructs within a multi-agent framework to enable team simulation capabilities. Eight additional characteristics of individuals and teams, chosen from a review of the design and psychology literature, are then layered on top of the multi-agent framework. It is shown that CISAT is capable of emulating the performance and behavioral characteristics of engineering design teams. In **Chapter 4**, CISAT is used to explore the relationship between the properties of design problems and the best team characteristics for solving those problems. Specifically CISAT is used to simulate team performance for different combinations of team size and interaction frequency on a variety of different configuration design problems. Regression analysis is then used to create relationships that enable the prediction of optimal team characteristics based on problem properties. In **Chapter 5** the results of a behavioral study are presented in which the effect of different interaction frequencies on the performance of human teams was examined. This study offers limited validation of the predictive relationships defined in Chapter 4, and demonstrates how the predictive equations can be used to offer new insights about how teams engage in the search for solutions.

Chapter 6 demonstrates the versatility of CISAT by using it as a starting point for the construction of a new, team-inspired numerical optimization algorithm. The core activity of agents in the CISAT framework is based on simulated annealing, a well-known class of optimization algorithms. Therefore, by stripping away all but the most quintessential team characteristics from CISAT, the Heterogeneous Simulated Annealing Teams (HSAT) algorithm is created. It is demonstrated the HSAT displays robust performance across a number of challenging objective function topographies.

Chapter 7 investigates the existence and effect of operation sequencing in engineering design. An analysis of data from cognitive studies shows that engineers employ strongly sequential operation patterns during design, and that these sequences can be modeled with first-order Markov Chains. CISAT is then used to conduct simulations demonstrating that designers who employ operation sequences are likely to generate solution with higher quality than those who do not use sequential patterns. Finally, hidden Markov models are used to evoke higher-level strategic sequences from the operational data.

In conclusion, **Chapter 8** summarizes results and key contributions from this dissertation. Potentially fruitful areas of future work are also identified and discussed.

Chapter 2: A Motivating Example: Comparing High- and Low-Performing Teams[1]

The beginning of an acquaintance whether with persons or things is to get a definite outline of our ignorance.

Mary Ann Evans

2.1 Overview

This chapter shares the results from a cognitive study that served to motivate the thesis of this dissertation. In the cognitive study, participants worked together in small teams to solve a truss design problem. It was demonstrated, firstly, that the highest-performing teams displayed low divergence. This is counter to literature stating that divergence is beneficial, motivating an exploration of how problem properties impact optimal team processes. Secondly, the results of the study showed that high- and low-performing teams not only differed in the characteristics of their solutions, but also in the processes that they engaged in to reach those solutions. This demonstrates that team characteristics can have a profound impact on solution quality, and need to be chosen in an informed manner.

The cognitive study in this chapter specifically explored the often-dynamic nature of the design process, which manifests through unexpected changes in goals or constraints. For

[1] This chapter is based on:

McComb, C., Cagan, J., and Kotovsky, K., 2015, "Rolling with the punches: An examination of team performance in a design task subject to drastic changes," Design Studies, **36**, pp. 99–121.

instance, a client might drastically change a set of specifications after solving begins, or a competitor may introduce a new technology or feature. Such unexpected changes are likely to require the team to perform some amount of redesign, ultimately decreasing the overall efficiency of the design process. Thus, the guiding question that drove the research in this chapter was: "How does a design team respond to drastic changes in the design task, and how can a team be made more resilient to these changes?" To answer this guiding question, two hypotheses were tested. First, we hypothesized that teams that excelled in responding to change would display underlying problem-solving processes that differed from teams that responded slowly or poorly. To explore these differences, a cognitive study was designed that tasked small teams of undergraduate engineering students with the design of a truss structure. Midway through the study, a fundamental aspect of the original design problem was changed. Shortly thereafter, a second modification was made to the original design problem. A complete record of the design team's efforts was collected through a computer interface, allowing problem-solving strategies to be fully reconstructed for analysis. Our second hypothesis was that considering additional design scenarios early in the design process would prepare teams to better respond to change. The results of the second hypothesis are not discussed here, but are available in the paper that this chapter is based on [36].

2.2 Background

For the purposes of this work, a change is considered to be drastic if the post-change problem requires a mental model or representation that is substantially different from that associated with the pre-change problem. In responding to a drastic change, a team must potentially overcome a variety of obstacles. One such obstacle is design fixation, defined as premature adherence to a design concept that impairs conceptual design efforts [37]. Fixation is relevant to this work because bias towards past solutions can be detrimental when responding to a change. This is particularly true if the problem is changed in such a way that drastically different solutions are required. A second obstacle is the effort required to simply become acquainted with the new problem representation. Still another is the selection of an adequate representation of the new problem, which must be done on-the-fly in a dynamic problem. Selection of a new representation impacts the extent to which knowledge can be transferred from the initial problem [38].

We predicted that high- and low-performing teams would display different problem-solving processes. Such procedural differences could result from the inherent variability of the individuals composing the team. The role of individual traits in addressing unexpected change at the team level was explored in several studies [39–41]. It was demonstrated that individuals' cognitive ability, goal orientation and openness to change are critical factors in predicting post-change performance [39–41]. Expertise is another phenomenon that can affect performance in engineering design, and leads to different solution strategies [42]. Since expertise may take years to develop [43], it is unlikely that *true* expertise was encountered in this work. However, students generally display varying levels of familiarity and experience on any given subject. Therefore, individual-level domain experience could lead some individuals to perform more like experts than others, inducing team-level differences. The work presented in this chapter randomly assigned individuals to design teams, making no attempt to control for such factors to homogenize the teams. Therefore, between-person variability was expected to induce a wide variety of problem-solving strategies at the team level.

In much of the work referenced above, as well as work by Fu, Cagan, and Kotovsky [44] and Dong, Hill, and Agogino [45], design quality was assessed by human evaluators. This procedure is both tedious for evaluators, and prone to human error. Additionally, both Fu, Cagan, and Kotovsky [44] and Dong and Agogino [46] used reports written by participants to evaluate convergence of the team on a common representation or problem solution. Convergence was also measured in this work, but design progress was directly recorded through a graphical user interface (GUI). This provided continuous design data, and makes written reports unnecessary. The direct recording of designs also enabled the objective evaluation of design quality.

2.3 Methodology

2.3.1 Experimental Overview

The primary purpose of this cognitive study was to examine how teams of engineers respond to drastic changes in their design task. Participants were assigned to teams of three, and given a truss design task, which was subjected to two substantial changes over the course of the study. Design was facilitated through the GUI shown in Figure 2.1. This GUI was written in MATLAB and allowed participants to build and test truss designs, as well as directly share designs within their team. All actions performed in the GUI were recorded for later analysis and reconstruction of solving efforts.

In addition to sharing designs through the GUI, participants were prompted to converse within their teams throughout the study. To further encourage collaboration within teams, participants were required at regular intervals to select and share their team's current best design. This ensured a minimum level of interaction, although all teams were observed to exceed this minimum.

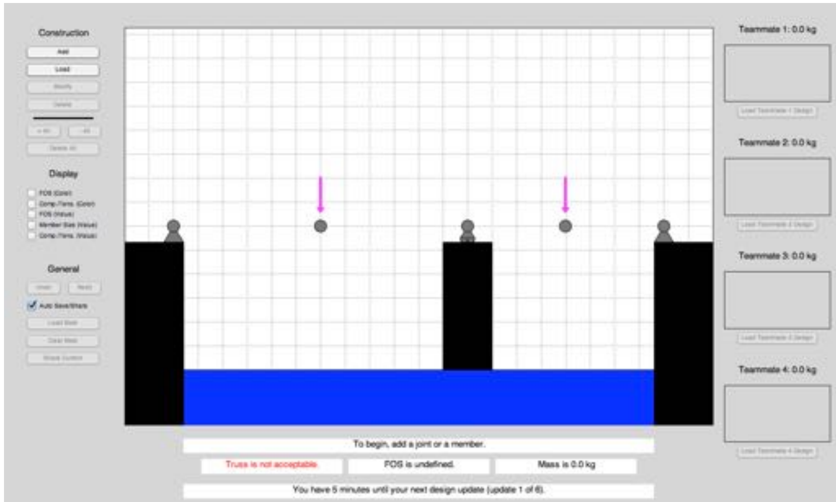


FIGURE 2.1. TRUSS DESIGN GUI, SHOWING LOADS AND SUPPORT.

2.3.2 Participants

This study was conducted with students in mechanical engineering senior design courses at two universities. Although it was unlikely these students had taken structural design courses, they were exposed to courses in statics and mechanics of materials in prior semesters. This ensured fluency in the basic knowledge required to understand the truss design task used in the study. In total, 48 students participated in teams of three (16 teams total) over the course of approximately one hour. Students were given course credit for their participation.

2.3.3 Materials and Design

Each participant was provided with access to a computer that was loaded with the truss design GUI and a tutorial program. The truss design GUI provided participants with the ability to build, evaluate, modify and share truss designs within their team. Shared designs were displayed to all members of the team as a thumbnail on the right side of the GUI window. The shared design could be imported directly into the workspace at the click of a button. Within the GUI, participants could apply loads of three different magnitudes (designated small, medium and large). The tutorial program provided an interactive experience that introduced participants to the truss design GUI.

The initial problem statement (PS1) provided to the teams was as follows:

1. Design a bridge that spans the river, supports a medium load at the middle of each span and has a factor of safety greater than 1.25.
2. Achieve a mass that is as low as possible, preferably less than 175 kg.

The meaning of “medium load” was explained to the participants through the tutorial program. Figure 2.1 depicts the loads and supports required for PS1. These remained the same for the second and third problems unless noted otherwise.

Over the course of the study the design problem was changed twice through the introduction of modified problem statements. The first modification required participants to consider the removal of any one of the bridge supports (leaving only two supports intact). The objectives in this problem statement (PS2) were stated as follows:

1. Design a bridge that spans the river, supports a medium load at the middle of each span, and has a factor of safety greater than 1.25.
2. Ensure that the bridge has a factor of safety greater than 1.00 even if any one support is destroyed.
3. Achieve a mass that is as low as possible, preferably less than 350 kg.

The second modification designated an area in which teams were not allowed to place structural elements (see Figure 2.2). The objectives for this problem statement (PS3) were stated as follow:

1. Design a bridge that spans the river, supports a medium load at the middle of each span, and has a factor of safety greater than 1.25.

2. Ensure that the bridge does not overlap or pass through the orange region.
3. Achieve a mass that is as low as possible, preferably less than 200 kg.

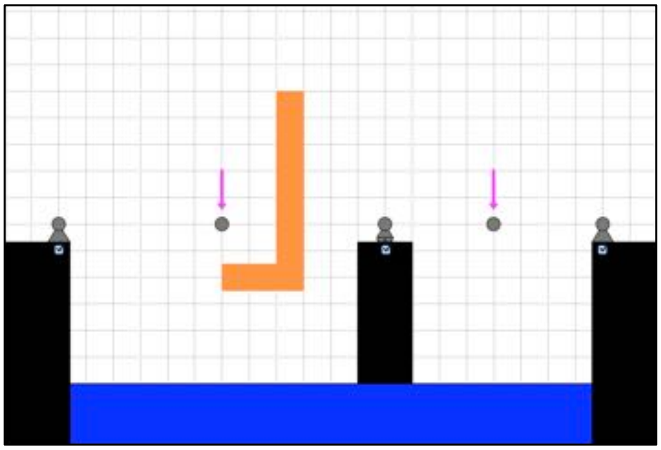


FIGURE 2.2. DESIGNATED AREA REFERRED TO IN PROBLEM STATEMENT 3.

2.3.4 Procedure

The study took approximately one hour, and a diagram of the time allocation is provided in Figure 2.3. Participants started with a 10 minute automated tutorial, completed individually. They were then given their initial problem statements, and instructed to discuss the problem statement within their team. After 5 minutes of team discussion, design commenced. Design was performed over the course of 6 periods, each 4 minutes in length. These design periods were separated by 1-minute interludes during which teams were prompted to select their current best design. Modified problem statements (PS2 and PS3) were provided after design periods 3 and 4, respectively. Participants were given 1 minute to read the new problem statements before being allowed to continue design.

The problem statement was changed twice (rather than once) in order to create a situation that emphasized ongoing flexibility and responsiveness. For the same reason, only one design session was allotted to PS2. This scheduled the changes back-to-back, placing further emphasis on the ongoing nature of a changing situation.

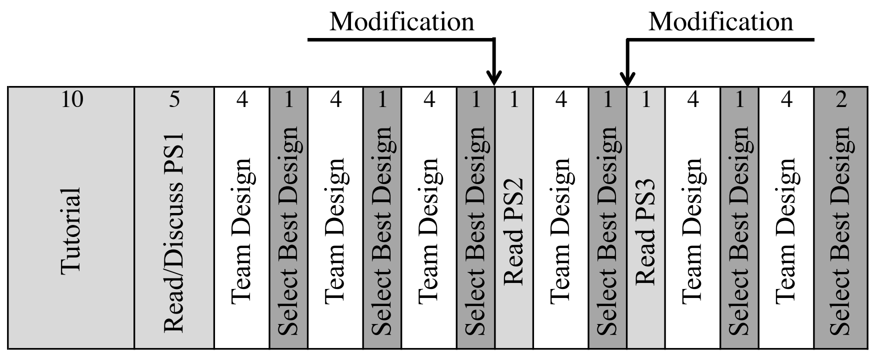


FIGURE 2.3. TIME ALLOCATION DURING TRUSS DESIGN STUDY WITH NUMBERS INDICATING DURATION IN MINUTES.

Although participants had access to individual computers, they were prompted to verbally interact with their team throughout the experiment. Additionally, the GUI allowed participants to directly share designs within their teams. This facilitated teamwork by allowing participants to adopt the designs of their teammates. Data was recorded whenever a participant modified their truss design, and whenever a truss design was shared. This allowed for later reconstruction of all moves within the design space.

2.4 Analysis

2.4.1 Quality Assessment

All problem statements contained two primary requirements: to minimize mass, and to achieve at least a minimum factor of safety (FOS). The requirement to minimize mass served as a goal for the problem, while the requirement to achieve a minimum FOS served as a problem constraint. The FOS is the primary measure of structural quality of a given design. To analyze the FOS for designs produced in this study, axial and buckling stresses were calculated using standard structural analysis techniques [47,48], and the FOS for a given structural member was computed as the ratio of allowable stress to the maximum computed stress. For PS1, the FOS was calculated for only a single support condition. However, Problem Statement 2 required participants to consider a variety of support conditions. To evaluate the quality for this problem statement, all support conditions were evaluated separately. The FOS of the design was then taken as the minimum FOS across all support conditions. For Problem Statement 3, the FOS was 0 if the region shown in Figure 2.3 was violated. Otherwise, the FOS was identical to that calculated for Problem Statement 1.

Portions of the analysis involved tracking the best design produced by a team over the course of the study. To respect the nature of the goal function (the mass of the solution) and the inequality constraint (the FOS of the solution) when inferring relative quality, the following guidelines were used to track the best design:

- If a design did not meet the FOS requirement, then another design was considered better if it had a higher FOS than the current design.
- If a design did meet the FOS requirement, then another design was considered better if it had a lower mass than the current design.

In other words, if the FOS constraint was violated, a design was considered to be better if it decreased the violation. If the FOS constraint was satisfied, a design was considered to be better if it decreased the mass.

The term *problem design solution* will be used to refer to the best design produced in response to any of the three problem statements. The strength-to-weight ratio (SWR) was used to analyze and communicate the results of the cognitive study. Note that teams were not shown the SWR of their designs during the study, and thus were not attempting to maximize SWR; rather, they were given the task of minimizing mass subject to a constraint on FOS. The SWR was used to analyze the results primarily because it allowed the mass and FOS to be communicated in an efficient and combined manner. The SWR was normalized according to the target FOS (FOS_T) and the target mass (M_T), as stated in the relevant problem statement, and calculated as

$$SWR = \left(\frac{FOS}{FOS_T}\right)\left(\frac{M}{M_T}\right)^{-1}. \tag{2.1}$$

A SWR greater than 1.0 indicated a structure that was relatively strong for its weight; a SWR less than 1.0 indicated that the structure was relatively weak for its weight. For some structures, the SWR was not an accurate indicator of structural quality. For instance, a truss with a low FOS and a very low mass could return a large SWR. However, when the best design was tracked as described above, such designs were surpassed by more valid solutions to the design problem.

Some analysis involved comparison of high- and low-performing teams. Teams were assigned to these two groups based on their total performance score. To calculate this score, teams were first scored in each of six sub-categories. These categories were the mass and FOS for the best design produced by the team in response to each of the three problem statements. Within each category, teams were assigned a score of $n - r + 1$, where n is the total number of teams (16), and r is the rank of the team in the category. The sum of a team's scores in each category yielded that team's total performance score. The 5 teams with the highest total performance scores were grouped together as the high-performing teams. Similarly, the 5 teams with the lowest total performance scores were grouped together as the low-performing teams.

2.4.2 Design Distance Measurement

In order to measure teams' exploration within the design space, it was first necessary to develop a quantitative method for comparing truss designs. To accomplish this, we defined operational distance to be number of joint and member operations needed to change one truss design topology into another. This was a relevant measure for assessing dissimilarity between designs produced during the study because operational distance counted the fundamental operations that the participants themselves used to traverse the design space.

Specifically, the operational distance was the minimum number of operations needed to construct the topology of the truss with more joints (denoted by Truss A) from the truss with fewer joints (denoted by Truss B). Here, topology is used to refer to the interconnection of the joints with structural members. The total number of operations was the sum of joint additions, member additions, and member deletions. Considering joint deletions was not relevant because Truss A had more joints than Truss B by definition.

The number of joint additions, *J*, was calculated as the difference in the number of joints between Truss A and Truss B. Determining the number of member additions and deletions was slightly more involved. First, the adjacency matrix of each truss was calculated. The adjacency matrices were *N*-by-*N* matrices (where *N* is the number of joints in Truss A). In the adjacency matrices, element *i,j* was 1 if a structural member connected joints *i* and *j*, and 0 otherwise.

The number of necessary member additions and deletions was calculated by minimizing the difference between the adjacency matrices of Truss A and Truss B. The difference, *D*, was calculated as

$$D = \sum_{i=1}^{N} \sum_{j=1}^{N} (A_{ij} \oplus B_{ij}), \qquad 2.2$$

where *A* and *B* are the adjacency matrices for Truss A and Truss B, respectively, and $\oplus$ is the XOR operator. The minimization of *D* was accomplished by reordering rows and columns in the adjacency matrix of Truss B to minimize the value of *D*. The physical interpretation was that of matching up joints between the two trusses to minimize the number of members that must be added or deleted to make Truss B and Truss A topologically identical.

As framed here, the operational distance minimization problem was an NP-hard combinatorial optimization problem. This precluded the calculation of exact solutions in a timely manner for all cases [49]. For operational distance calculations involving only small trusses it was feasible to compute an exact solution. However, for calculations involving larger trusses a stochastic, greedy algorithm was utilized. Once *D* was minimized, the operational distance, d_{op}, between Truss A and Truss B was calculated as

$$d_{op} = J + D/2. \tag{2.3}$$

The value of the difference, *D*, was divided by two to account for the symmetric nature of the adjacency matrices.

2.4.3 Distance-Related Assessments

Using the concept of operational distance, a number of distance-related assessments were defined to yield insight into team problem-solving processes.

The first, *average pairwise distance*, was the distance between the designs being explored by any two members in a team at a given instant, averaged across all combinations of two team members. A similar notion, referred to as average pairwise similarity, has previously been used as an indicator of convergence, or agreement on a common solution concept [44,50,51]. Conversely, we took average pairwise distance to be an indicator of divergence, or disagreement on a common solution concept.

Distance to problem design solution referred to the operational distance between the designs currently being explored and the problem design solution eventually produced for the current problem statement. This metric was used to communicate important characteristics with respect to how a team approaches its solutions.

Rate of exploration was the number of operations per minute. This was calculated for each participant by sampling their current design every two minutes. The rate was then calculated by dividing the operational distance between consecutive designs by two minutes. This yielded a rate measurement for every half of a design session. This conveyed a sense of how quickly participants traversed the design space.

2.5 Results

2.5.1 Performance Results

The 16 teams were sorted according to their total performance score. In order to assess patterns that were associated with successful teams, the 5 highest-performing teams were compared to the 5 lowest-performing teams. A plot of the average quality of the best designs produced by these two groups is provided in Figure 2.4. The best design was tracked within each team using the bifurcated relative quality guidelines introduced previously. Vertical gray bars indicate periods during which teams were instructed to stop designing and select a current best design. Vertical gray bars overlaid with the text "Change" indicate a break during which a new problem statement was provided to the team. This new problem statement (PS2 or PS3) modified central characteristics of the previous problem statement.

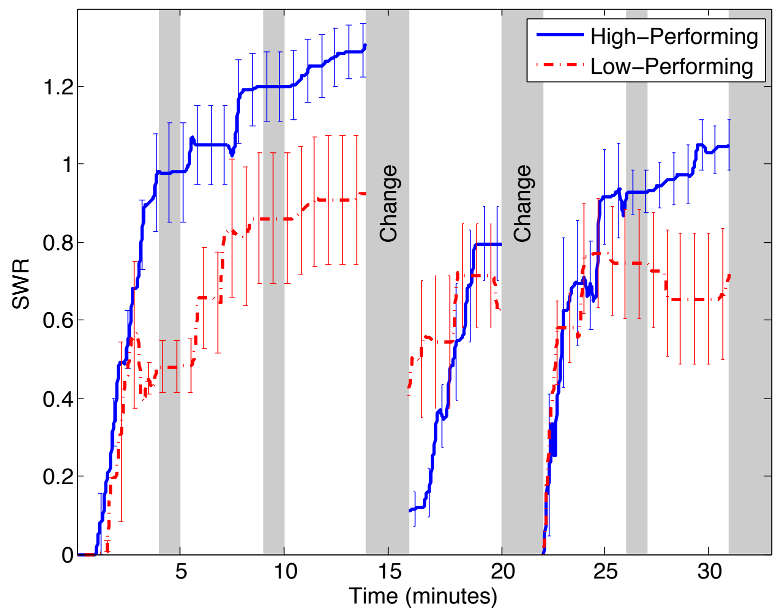


FIGURE 2.4. STRENGTH-TO-WEIGHT RATIO OF BEST DESIGN FOR HIGH- AND LOW-PERFORMING TEAMS. ERROR BARS SHOW ±1 S.E.

The problem design solutions of the high-performing teams were generally better, and never worse with one exception, to those of the low-performing teams throughout the study. There was a considerable difference in the SWR of the problem design solutions produced for PS1 (at 14 minutes). For PS2, the difference in SWR at the end of that problem session was not significant. However, the extent to which the high-performing teams increased the SWR of their best design between 16 and 20 minutes during PS2 was substantial. Interestingly, the very start of PS2 was the only place that low-performing teams briefly performed better than the high-performing teams but the high-performing teams quickly recovered. While working on PS3, both high- and low-performing teams initially recovered quickly. However, the low-performing teams soon stalled, and were overtaken by the high-performing teams.

Further, high-performing teams also showed a pattern of divergence that was quite different from the low-performing teams. Figure 2.5 shows the average pairwise distance through the study.

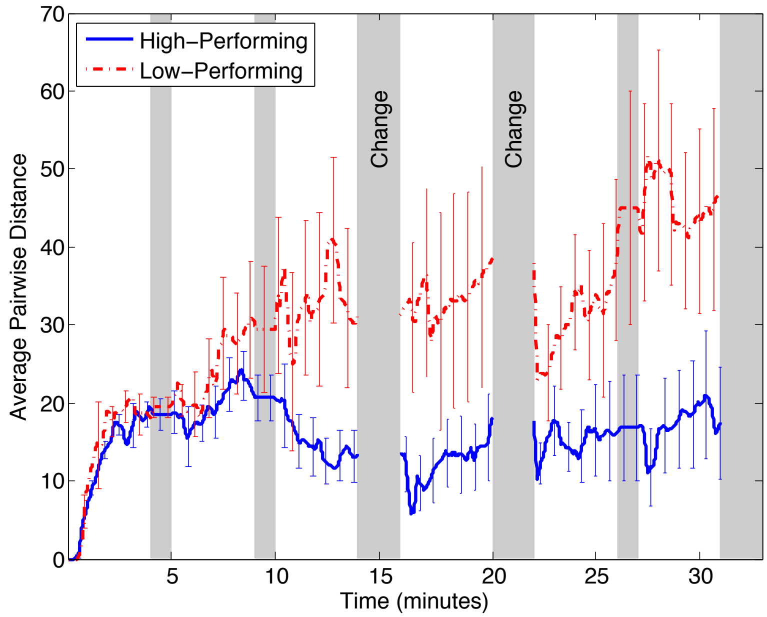


FIGURE 2.5. AVERAGE PAIRWISE DISTANCE OF HIGH- AND LOW-PERFORMING TEAMS. ERROR BARS SHOW ±1 S.E.

Both high- and low-performing teams diverged quickly at the beginning of the study. However, the high-performing teams soon began to converge, and maintained a roughly steady level of lower divergence for the remainder of the study. On the other hand, the low-performing teams continued to diverge, and maintained a relatively high level of divergence through the remainder of the study.

Examining the distance to problem design solution reveals further differences between high- and low-performing teams. Figure 2.6 shows the distance to problem design solution, averaged over all individuals. Figure 2.7 shows a similar plot that only includes the closest member in each team. Figure 2.6 indicates that the low-performing teams made progress towards and then *away from* their problem design solution. This trend is echoed more strongly in Figure 2.6. The closest members in the low-performing teams did get close to the best solution, but were soon pulled away by the majority influence of the rest of the team. On the other hand, the closest members of the best-performing teams made quick progress towards the solution, and then remained in close proximity.

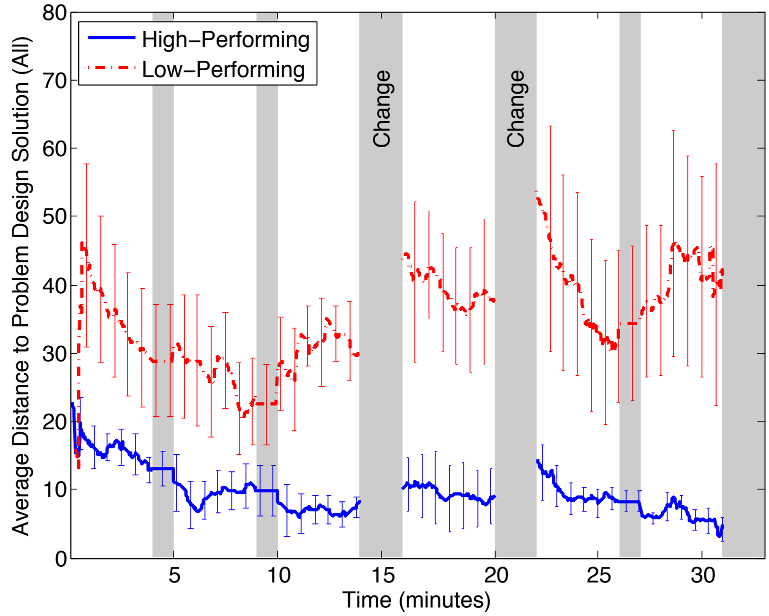


FIGURE 2.6. AVERAGE DISTANCE TO PROBLEM DESIGN SOLUTION FOR ALL PARTICIPANTS. ERROR BARS SHOW ±1 S.E.

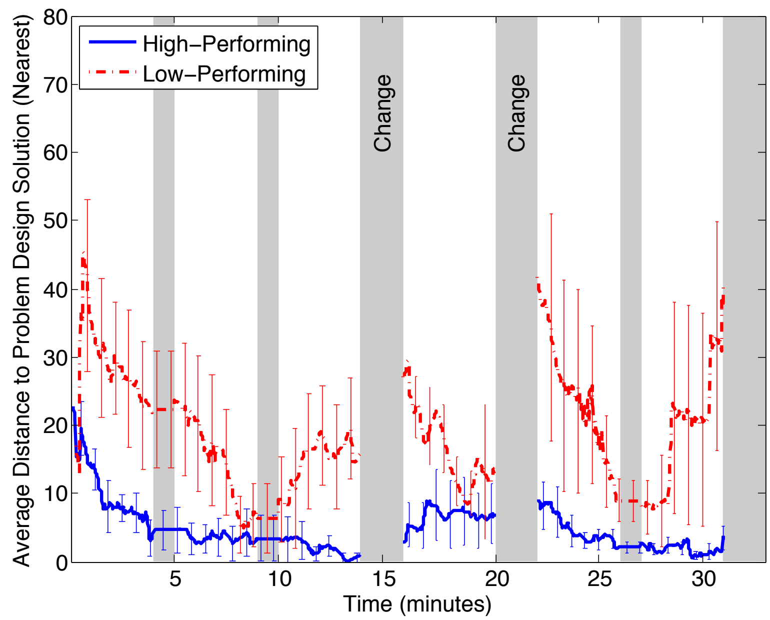


FIGURE 2.7. AVERAGE DISTANCE TO PROBLEM DESIGN SOLUTION FOR NEAREST PARTICIPANT FROM EACH TEAM. ERROR BARS SHOW ±1 S.E.

With the delivery of a new problem statement, the low-performing teams began working at a very large average distance from their problem design solution. On the other hand, the high-performing teams were never very far from their problem design solution during work on any of the problem statements. This implies that the high-performing teams produced similar solutions for the three problem statements, while the low-performing teams changed their solutions considerably over time. The average distance between problem design solutions produced within a team is shown in Figure 2.8.

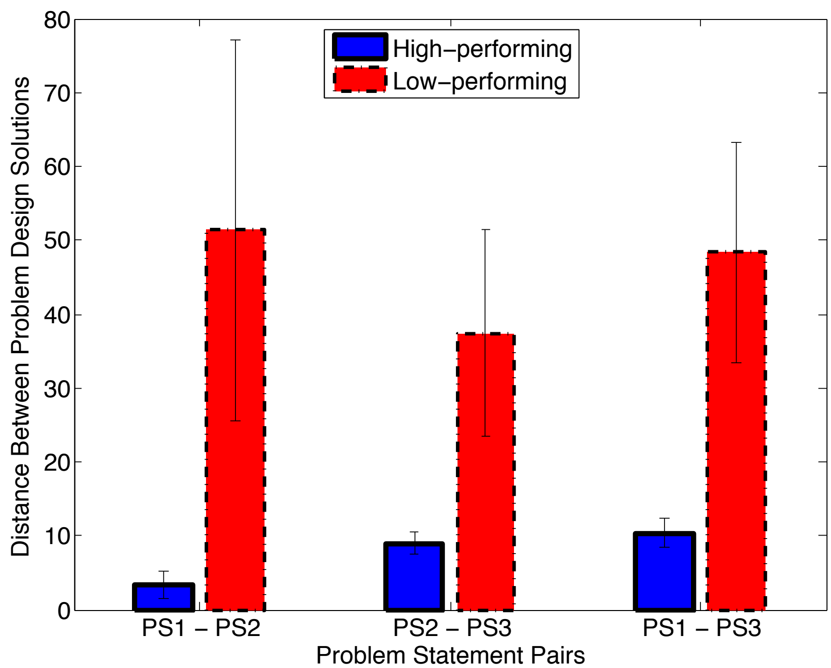


FIGURE 2.8. AVERAGE DISTANCE BETWEEN PROBLEM DESIGN SOLUTIONS PRODUCED WITHIN A TEAM. ERROR BARS SHOW ±1 S.E.

Given the similarity between consecutive solutions in the high-performing teams, it appears that their exploration of the design space was limited. This was, however, not the case. Figure 2.9 shows the average rate of exploration. An "A" affixed to the design session number indicates the first half of the session, and a "B" indicates the second half of the session. For instance, a label of "3A" on the abscissa refers to the first half of design session 3.

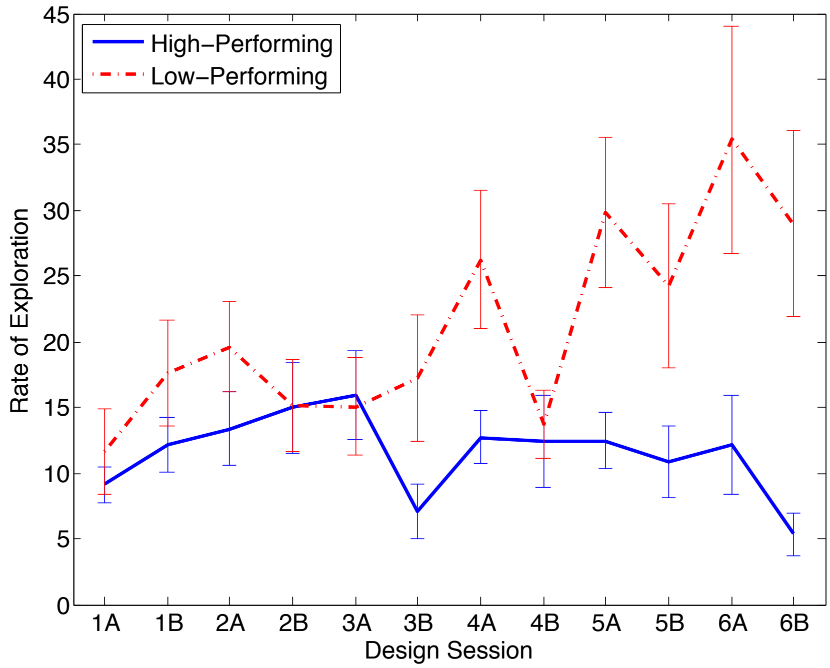


FIGURE 2.9. AVERAGE RATE OF EXPLORATION. ERROR BARS SHOW ±1 S.E.

Figure 2.9 indicates that individuals in both high- and low-performing teams initially traversed the design space at nearly the same rate. However, after the first change, individuals in the low-performing teams began to explore the space much more quickly. The rate continued to increase through the end of the study. Although Figure 2.9 shows that both high- and low-performing teams explored at comparable rates during early stages of design, team performance in Figures 2.6 and 2.7 is distinctly different. For the high-performing teams, the combination of moderate search rate and close proximity to their eventual solution indicates that they actively explored a very localized portion of the design space. Members of the low-performing teams, on the other hand, exhibited a similar search rate with a large initial distance from their eventual solution. This indicates that they expended less effort in local exploration, and spent more effort in traversing the design space. Similar evidence is found when comparing Figures 2.8 and 2.9.

While working on PS2 (design session 4), the rate of exploration of high- and low-performing teams was similar in magnitude. However, the distance between problem design solutions for PS1 and PS2 is disproportionately lower for high-performing teams. This indicates that they explored a relatively small part of the design space.

For the purposes of this cognitive study, design complexity was proportional to the number of structural elements. Information regarding the number of joints and members in problem design solutions is provided in Figure 2.10.

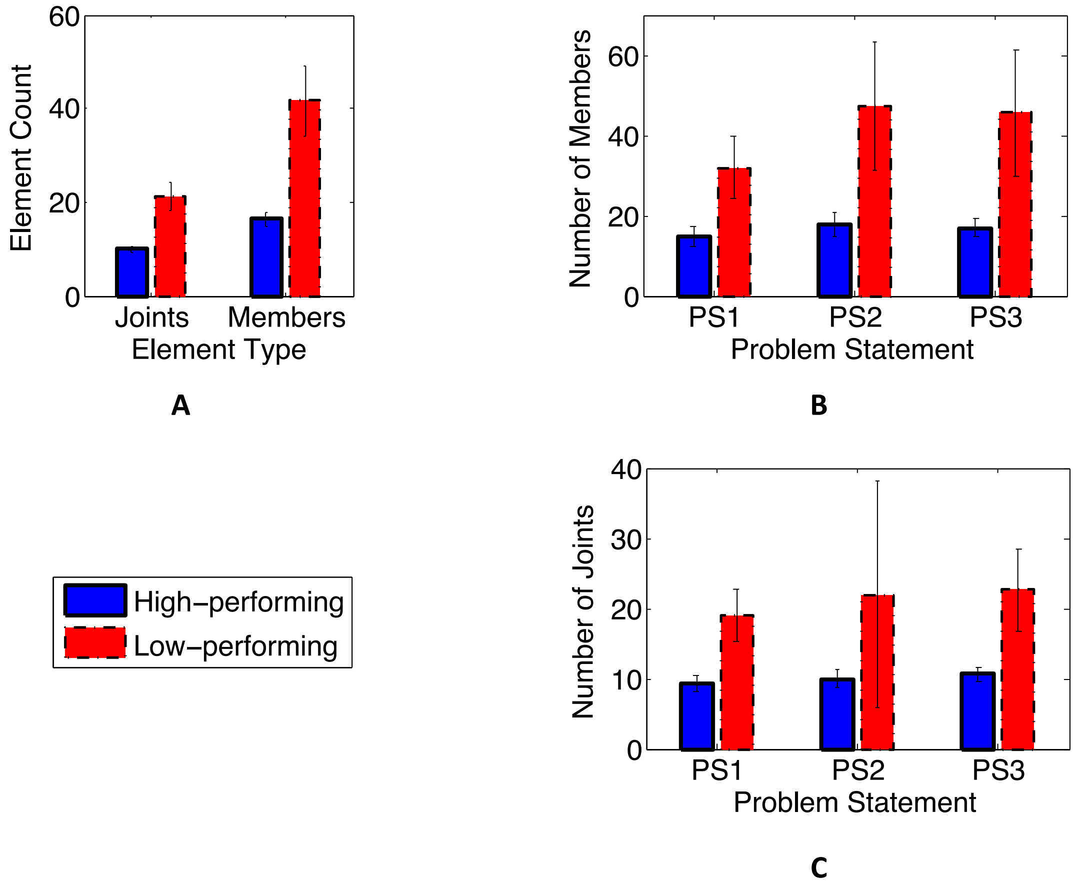


FIGURE 2.10. COMPLEXITY OF PROBLEM DESIGN SOLUTIONS SHOWING (A) ELEMENT COUNT AVERAGE ACROSS ALL PROBLEM STATEMENTS, (B) NUMBER OF MEMBERS BY PROBLEM STATEMENT AND (C) NUMBER OF JOINTS BY PROBLEM STATEMENT. ERROR BARS SHOW ±1 S.E.

Figure 2.10.a shows the average complexity of all problem design solutions, indicating that low-performing teams created problem design solutions with much greater complexity. In addition, Figures 2.10.b and 2.10.c show the number of joints and members of the problem design solution produced for each problem statement. High-performing teams maintained a steady level of complexity across all problem design solutions. The low-performing teams displayed a notable increase in the number of structural members between PS1 and PS2, but did not rebound to a lower number as they produced a solution to PS3. Representative problem design solutions for PS2 are provided in Figure 2.11. This serves to illustrate the trends displayed in Figure 2.10.

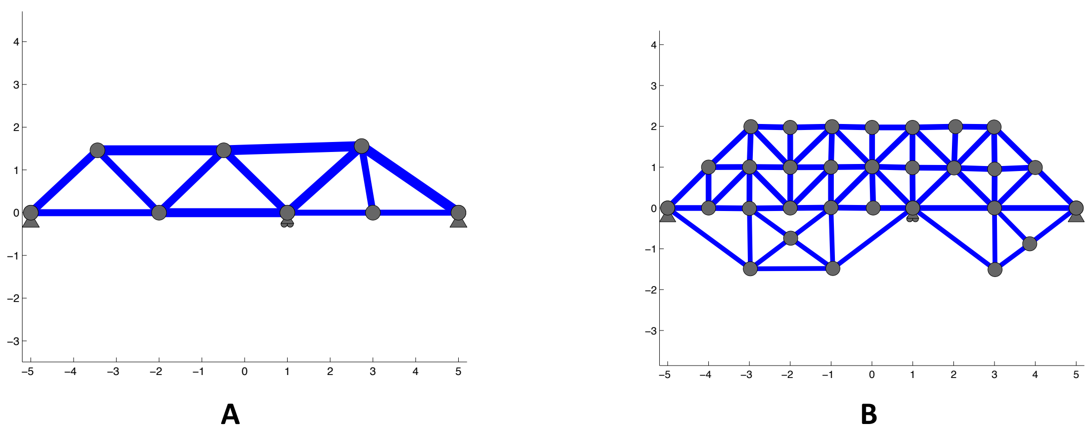


FIGURE 2.11. REPRESENTATIVE SOLUTIONS TO PROBLEM STATEMENT 2 FOR (A) HIGH-PERFORMING TEAM AND (B) A LOW-PERFORMING TEAM.

A summary of the principal differences between high- and low-performing teams in this study is provided in Table 2.1.

TABLE 2.1. SUMMARY OF DIFFERENCES BETWEEN HIGH- AND LOW-PERFORMING TEAMS

Metric	Relevant Figures	High-performing	Low-Performing
Divergence	Figure 2.5	Low; increasing, then decreasing	High; increasing
Average distance to solution	Figures 2.6, 2.7	Low; decreasing	High; decreasing, then increasing
Distance between problem design solutions	Figure 2.8	Low	High
Average Rate of Exploration	Figure 2.9	Fairly constant	Increasing
Problem design solution complexity	Figures 2.10, 2.11	Low	High

2.6 Discussion

The first hypothesis of this chapter proposed that teams that excelled in responding to change would display underlying problem-solving processes that differed from teams that responded slowly or poorly. High- and low-performing teams *did* display very different problem-solving patterns, thus confirming this hypothesis. Notably, high-performing teams tended to display high convergence, arrived at relatively simple problem design solutions, and searched specific/small areas of the design space. On the other hand, low-performing teams displayed high divergence, tended to develop complicated designs, and did not search specific targeted areas.

Goal orientation theory offers one possible explanation for some of the observed differences. Work by LePine [41] indicated that goal orientation plays a crucial role in how teams adapt to change. Goal orientation describes how individuals respond when placed in an achievement setting [52]. Performance goal orientation indicates a desire to avoid failure and receive favorable judgment, while learning goal orientation places an emphasis on achieving mastery of the task [53]. Individuals displaying learning goal orientation tend to respond to negative feedback constructively, while those with a performance goal orientation are unlikely to learn from negative feedback [53]. Teams in which individuals display learning goal orientation adapt more easily to change [41].

In the study presented here, both high- and low-performing teams produced early designs with very similar quality (see Figure 2.4). After approximately 3 minutes, the low performing teams stalled, while the high-performing teams continued to improve. While improving, they searched a small portion of the design space. A possible interpretation for this behavior is that the high-performing teams sought mastery of this small portion of the space, which could be indicative of learning goal orientation. The low-performing teams, on the other hand, continued to traverse the design space when their early attempts demonstrated low quality.

High- and low-performing teams produced designs that differed substantially in terms of complexity. A possible explanation for this behavior can be provided by cognitive load theory. We propose that the simplicity of the designs produced by high-performing teams actually assisted them in understanding and learning about the space, which in turn enabled them to respond readily to changes. Cognitive load theory states that the cognitive load associated with a task is composed of intrinsic load (the difficulty that is inextricably linked with the task), extraneous load (the additional load generated by the method in which material is delivered), and germane load (the load devoted to processing information and constructing schemas) [54]. If the cognitive load associated with a task exceeds an individual's available working memory, meaningful learning may not occur [55].

Through producing simple designs, the high-performing teams placed little extraneous cognitive load on themselves. This allowed remaining working memory to be devoted to schema construction, which allowed members of these teams to learn how to reason within the design space. Their ability to learn quickly and effectively assisted them in responding to changes delivered during the course of solving. Members in low-performing teams tended towards more complex designs, which imposed a far greater extraneous load. Therefore, these individuals had less working memory to apply towards schema formation, which inhibited their ability to learn and reason within the design space. This impacted their ability to respond quickly and precisely to changes. Similar results regarding the interaction between problem representation, cognitive load and problem-solving difficulty have been demonstrated in other domains [56,57].

Interpreting the results in terms of expertise offers another possible explanation for the observed differences. Here, we define expertise as a high level of performance in a given task [58], acquired through a sustained and deliberate period of effort [43]. It has been demonstrated that problem-domain experts can quickly and accurately classify problems, and begin moving more or less directly towards a solution [59]. Specifically in the context of design, expert designers tend to quickly commit to a single solution concept, rather than exploring a variety of alternatives [42]. However, true expertise is generally the result of years of effort [43], so it is very unlikely that any participants in this study were truss design experts. However, individuals in high-performing teams behaved in some ways like experts, quickly selecting a good direction in which to search. In addition, they exhibited a low level of divergence, similar to the balanced search strategies observed in expert designers [60]. Generally, the strategies observed in this study's high-performing teams suggest that they possessed relevant skills or knowledge that individuals in low-performing teams did not. The general strategy that these high-performing teams used to solve the problem has been identified. For design tasks similar to truss design, this strategy can be taught to teams in order to increase the likelihood of expert-like behavior.

The concept of team mental models also provides a framework with which to interpret the results of this study. Previous work has demonstrated that characteristics of team mental models can be strong predictors of team performance [61,62]. In the cognitive study presented here, participants were instructed to discuss PS1 within their team before beginning work. It is possible that some teams developed shared mental models of higher quality during this time, particularly with respect to the level of sharedness. A high quality team mental model could have enabled such teams to perform at a higher level, and also acted to limit divergent search, resulting in the lower divergence exhibited by high-performing teams. What of the low-performing teams? Typically, groups under stress (such as that brought about by the changes in this study), exhibit an increased desire for group consensus [63]. However, if individuals already have strongly held preferences, stress can induce an opposite response – a reduced willingness to acquiesce to others [64]. It is likely that individuals in low-performing teams developed distinct preferences over design alternatives, evidenced by high divergence. Therefore, these teams could have experienced a reduced willingness to acquiesce when encountering the changes, leading to a further increase in divergence. In turn, the general lack of consensus might have further exacerbated earlier low performance.

The problem-solving literature suggests that early divergence can be beneficial for problem-solving (for instance [65,66]). However, results from this chapter indicate that high-performing teams display low divergence throughout the solving process. While the problem explored in this chapter constrained teams to design a truss to span the river (as opposed to beam structures, or other potential solutions), many problems studied in the literature have more open-ended design spaces. Another difference is that many design problems used in the literature tend to be novel for the participants, while truss design is ubiquitous in mechanical engineering education, thus making it more familiar to the participants. These differences indicate that there may exist a range of problem types that correspond to a range of optimal solution strategies.

2.7 Summary

The analysis in this chapter demonstrated that high- and low-performing teams varied in the approaches that they employed to solve a design problem subject to two drastic changes. High-performing teams tended to display high convergence after a brief period of controlled divergence, and searched focused regions within the design space. In contrast, low-performing teams displayed an extended period of high divergence, and did not target specific areas of the design space. In addition, high-performing teams tended to arrive at simple final designs, whereas low-performing teams arrived at final designs with a higher degree of complexity. Plausible explanations for the observed differences between high- and low-performing teams were offered through variability in domain experience, goal orientation, and self-imposed cognitive load. In addition, the characteristics of solutions produced by high-performing teams support the potential efficacy of design simplicity as a strategy for responding quickly to change. However, this strategy may only be valid for problems similar in nature to truss design, and should be examined further.

The primary result from this study is that high-performing teams displayed relatively low divergence. This contrasts with several studies in the design literature that indicate the benefit of divergent search. The superiority of a low-divergence approach likely resulted from the properties of the task itself. Thus, the relationship between knowable properties of a given task and the most effective team characteristics for solving that task should be explored in greater detail. The explication of such a relationship will be the primary focus of this dissertation.

Chapter 3: The Cognitively-Inspired Simulated Annealing Teams Modeling Framework[2]

Ugly programs are like ugly suspension bridges: they're much more liable to collapse than pretty ones, because the way humans (especially engineer-humans) perceive beauty is intimately related to our ability to process and understand complexity.

Eric S. Raymond

3.1 Overview

Empirical studies are a common means for exploring design cognition and for testing new design methodologies. However, these studies can incur a high personnel cost while only returning a limited amount of data. It can also be difficult to isolate the effects of specific characteristics. This chapter introduces the Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework, a platform that simulates team-based engineering design through creating software agents that directly solve engineering problems. In addition to offering a resource efficient test bed for evaluating design strategies, this framework can be used to test the conclusions from cognitive studies. It can be used to peel apart aspects of human design, and provides a succinct representation of designer behavior. The purpose of the framework is not to replace cognitive

[2] This chapter is based on:

McComb, C., Cagan, J., and Kotovsky, K., 2015, "Lifting the Veil: Drawing insights about design teams from a cognitively-inspired computational model," Design Studies, **40**, pp. 119–142.

studies, but rather to augment traditional methods of investigation, accelerating the discovery of improved design methodologies.

The CISAT framework makes use of simulated annealing constructs to model several characteristics of individuals in small design teams. It differs from other simulation models because it strikes a balance between model simplicity and direct applicability, offering a succinct modeling framework that can be used to directly solve engineering design problems. This chapter first provides a description of the characteristics that are modeled within CISAT. Next, the CISAT framework is used to simulate the results of the cognitive study conducted in Chapter 2. These results are directly compared to results derived from human designers performing an identical task. Following this validation, the CISAT model is used to evaluate which characteristics were most and least helpful to teams during the cognitive study.

3.2 Background

The current chapter draws upon demonstrated phenomena in the design, problem-solving, and psychology literature in order to construct a computational representation of designer activity. To that end, this section reviews relevant literature on the behavior of human designers and alternative computational models for both individuals and teams.

3.2.1 Designer and Team Activity

When humans solve a problem, they tend to learn strategies that can be expressed in terms of the move operators that apply to the problem [67,68]. Solution strategies can also be expressed in terms of search breadth. Expert designers employ a mixture of different breadth- and depth-first search strategies during solving [69]. The selection of appropriate search strategies is further impacted by the presence of known goals or targets. It is known that individuals tend to satisfice, meaning that they only search the solution space until a solution that satisfies relevant targets is found [70]. It is not uncommon for designers to have direct knowledge of goals during the design process. For instance, the widely used target costing method determines goals before design begins, and these goals are used to guide the search for solutions throughout the design process [71,72].

Teams are composed of individuals who strive towards a common goal [73]. When members of a team interact while working towards the goal, they perform better than individuals working alone [50,74]. Interaction is usually observed to occur organically, taking place at irregular intervals [75]. The performance boost from interaction is caused by the ability of a team to initially diverge to explore a variety of options, but then converge at the right time, focusing the attention of the team members on a diminishing set of alternatives [44,45]. However, premature convergence on a single solution can be detrimental to solution quality [37]. For that reason, designers are typically taught to explore multiple solution concepts [65,76]. Even members of high-performing teams tend to pursue slightly different solution concepts when solving well-defined problems [36], indicating that members of a team don't always greedily pursue the solutions with highest apparent quality. Therefore, though team members may factor design quality into decisions, they freely pursue designs that may currently display lower quality. Interaction between members of a team is further tempered by preference for one's own designs. In particular, designers are known to largely favor their own designs, often preferring to apply numerous patches to early design concepts than explore alternatives [77,78]. Designers have also been shown to preferentially evaluate their own solution concepts [79,80].

3.2.2 Computational Models

A significant amount of work has attempted to simulate the performance of both teams and individuals [81]. For instance, both the Virtual Design Team model, and another model applied to teams at NASA's Jet Propulsion Laboratory, incorporate detailed descriptions of design team organization and interaction [27,28]. Both models were used to simulate complex design tasks, but were also burdened by high model complexity. For instance, the model by Olson et al. [28] used approximately 1000 distinct variables, and required nearly 100000 lines of code for implementation. Still other work has utilized agent-based models to explore the formation of mental models during team problem-solving with respect to both interaction structure [29] and agent memory [31]. Mental models were created by either adding noise to the true problem function or interpolating between known function values. That work obtained results that agreed qualitatively with the literature, but only explored one- and two-dimensional continuous problem domains, and was not compared to the results of any human studies. Other work used a multidisciplinary modeling framework to study team design, but was also limited through the use of a continuous domain design problem with a small number of design variables [82]. A recent agent-based design team model also explored the effect of team structure and task complexity on the formation of transactive memory [30,32]. That work also obtained results that agreed qualitatively with the literature, but modeled the design problem as an abstract task network instead of directly solving a concrete design problem. Other work simulated with great detail the tasks involved in an integrated product development team, but did not apply the model to a real design task, or offer empirical validation [83].

Simulated annealing [35] is a stochastic optimization algorithm that has been used to effectively model individual human problem-solvers [33] and designers [34]. The effectiveness of simulated annealing for this purpose might be explained by its close relationship to generative sensing [84], a pattern of designer activity. Other work has demonstrated the potential benefit of using computational agents to rapidly test and refine rule-based search strategies that can then be provided to human designers [85]. The rule-based search strategies included both stochastic and univariate approaches. In that work, both computational agents and human participants solved a continuous domain problem with a small number of variables, but the work was not extended to more complex problems.

3.3 The CISAT Modeling Framework

The CISAT modeling framework is an agent-based platform that is intended to simulate the process and performance of human design teams. A conceptual flowchart for the CISAT modeling framework is provided in Figure 3.1. Although only three agents are depicted in the flowchart, the framework is general and can model larger teams.

The CISAT framework models 8 characteristics that contribute to a description of how individuals solve problems, both independently and as part of a team. These characteristics are listed briefly below, and explained in greater detail in subsequent sections:

1. Multi-agency: A team is a collection of individuals with a common goal [73].
2. Organic interaction timing: Interaction within teams occurs at irregular intervals [75].
3. Quality-informed solution sharing: Members of a team tend to focus on the most promising alternatives, but don't do so greedily [36].
4. Quality bias reduction: Individuals in a team develop multiple solution concepts to avoid premature convergence [37,65,76].
5. Self-bias: Designers tend to be biased in favor of their own designs [77–80].

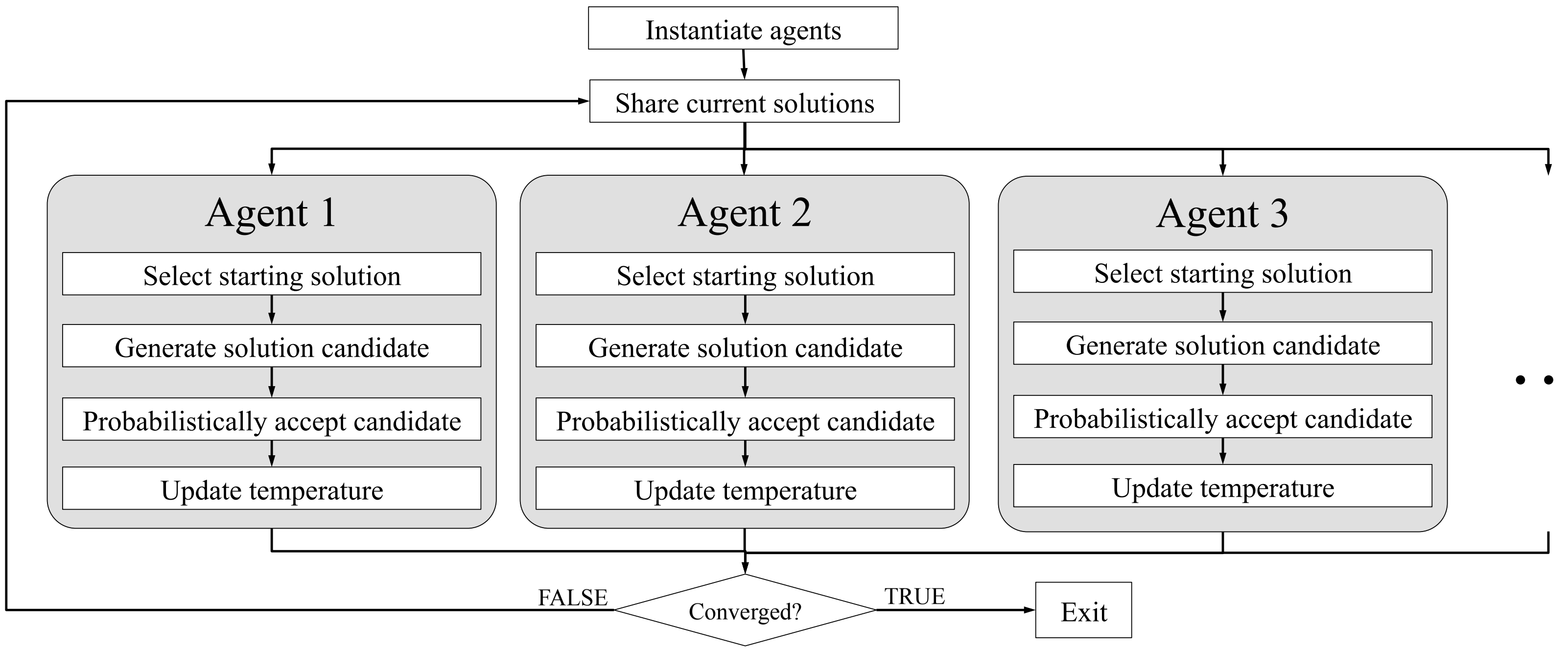


FIGURE 3.1. CONCEPTUAL FLOWCHART FOR THE COGNITIVELY-INSPIRED SIMULATED ANNEALING TEAMS (CISAT) MODELING FRAMEWORK.

6. Operational learning: Individuals learn strategies over the course of solving [68].
7. Locally Sensitive Search: Designers select from a range of breadth- and depth-first search strategies as they explore the design space [69].
8. Satisficing: Individuals only search the solution space until a solution is found that satisfies relevant targets [70].

Randomized selection is employed in several of the following sections. If the selection is made from a discrete set of alternatives, a multinomial distribution is used (which can be thought of as a roll of a weighted die). If the selection involves choosing a value from within some range, a uniform distribution is employed (equal probability for all values within the range).

3.3.1 Multi-Agency

The modeling framework is based upon collaboration between multiple software agents. A software agent, referred to simply as an agent in this work, is a computational routine that senses an environment and independently responds to that environment [86]. For CISAT agents, the environment is the problem space, and they sense it by evaluating potential solutions. Agents then respond by creating, sharing, and refining solution concepts. Within CISAT, every human designer is modeled by exactly one agent. These agents share a common goal (the minimization of an objective function) making them a suitable proxy for members of a team [73].

3.3.2 Organic Interaction Timing

The amount of inter-member communication varies between teams, and occurs at irregular intervals [75]. Similarly, interaction between agents in CISAT occurs probabilistically. Agents independently and probabilistically choose whether or not to interact at the beginning of every iteration. If an agent chooses not to interact, then it continues to iteratively modify its own design. If an agent chooses to interact, it selects a design to explore from amongst the design alternatives currently being pursued by the team (it may select its own design through this process). The selection probability is imperfectly informed by the relative quality of design alternatives, and adjusted to account for self-bias.

3.3.3 Quality-Informed Solution Sharing

Although a team is composed of individual problem-solvers, there is often additional benefit that is derived from interaction between the individuals [50]. This arises from the ability of individuals in a team to explore a variety of options, but also to collaboratively focus their attention on a shrinking set of the most promising alternatives [44,45]. Even members of high-performing teams tend to pursue slightly different solution concepts while solving well-defined problems [36], indicating that members of a team don't always greedily pursue the solutions with highest quality. Therefore, though team members may factor design quality into decisions, they freely pursue designs that may currently display lower quality.

The CISAT selection process attempts to model the above description of interaction by allowing agents to probabilistically choose to adopt the current design of any other agent in the team. The selection probability of a design is proportional to its weight, $\boldsymbol{W}$, which is defined as:

$$\boldsymbol{W} = -\boldsymbol{F} + \max(\boldsymbol{F}). \tag{3.1}$$

The vector $\boldsymbol{F}$ contains the objective function value for each design in the set of designs currently being pursued by the agents of the team. This equation makes the selection probability of each design proportional to its quality (relative to other available designs). Once the weighting vector, $\boldsymbol{W}$, has been computed, the agent selects a design alternative by choosing a design with probability proportional to its weight. This selection process can be visualized as the spin of a roulette wheel, or the roll of a loaded die. Once an agent has selected a design alternative to pursue using this probabilistic process, it proceeds to modify that design independently using an internal process structured similarly to simulated annealing.

3.3.4 Quality-bias Reduction

Note that in the weighting vector $\boldsymbol{W}$, the weight placed on the worst design is 0. This means that agents are incapable of selecting the worst design when interacting, and abandon it automatically. This detail could lead to premature convergence within the team, which can be harmful to design [37]. Although novice designers are often taught to explore multiple solution concepts [65,76], expert designers do not generally exhibit this behavior [42]. This implies the existence of a range of strategies that are employed along the spectrum from novice to expert. To imbue CISAT with the ability to accommodate this range, a small additional weight is added to $\boldsymbol{W}$:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} + k^{\text{QBR}} \cdot \mathbf{1}. \qquad 3.2$$

The variable k_{QBR} controls the strength of quality bias reduction, and $\mathbf{1}$ is the ones vector. This reduces the effect of the agents' bias towards designs of high quality. This also places non-zero weight on the worst design alternative, meaning that agents may select any solution concept.

3.3.5 Self-bias

Designers tend to be biased in favor of designs that they have generated or spent substantial time working on [77–80]. Therefore, CISAT agents are also made to favor their own designs. This is implemented during interaction between agents. Before an agent selects a design to pursue, the agent adds additional weight to the element in $\boldsymbol{W}$ that corresponds to its own design:

$$W_i \leftarrow W_i + k^{\text{SB}}. \qquad 3.3$$

The variable k_{SB} controls the strength of self-bias, and the subscript i denotes the index for the current design of the agent making the selection. This results in a higher likelihood that the agent will elect to continue working on its own design, mimicking the bias of human designers.

3.3.6 Operational Learning

The actions that can be used to modify a solution are typically referred to as *move operators* and are inherently problem specific [67]. Human problem-solvers tend to learn strategies in terms of these move operators [68]. CISAT agents are also provided with a mechanism that allows them to individually learn which move operators are most helpful to design.

Within the CISAT framework all move operators initially have an equal probability of being selected and applied to the current solution (unless a prior distribution over move operators can be inferred for the specific problem). Agents learn by first selecting a move operator, and using it to modify the current solution. The new solution is then evaluated using the objective function. If the application of the move operator improved the objective function, the probability of applying that move operator in the future is increased. However, if the application of the move operator gave the solution a worse objective function value, the probability of applying the move operator is decreased.

This method of tuning the selection weights of move operators is similar to other methods for learning which moves to apply such as rule-based interactive tree search [87,88] or the Hustin method [89], which has previously been applied to the design of truss structures [90]. The operational learning characteristic used here is significantly simpler than these alternative approaches, but is still capable of resolving a human-like transition from topology to parameter optimization (identified later in this chapter). The exact implementation of this characteristic is explained in greater detail in Chapter 7.

3.3.7 Locally Sensitive Search

It is known that expert designers tend to use a mixture of depth- and breadth-first solution strategies [69], indicating the value of tailoring search strategies to local characteristics of the design space. CISAT is based on a simulated annealing methodology, so the annealing schedule controls the progressive transition from initial explorative search to final deterministic search. To mimic the locally sensitive search strategies of human designers, every CISAT agent is given an independently-controlled Triki adaptive annealing schedule [91]. This annealing schedule uses the variance of the quality of past solutions to update the temperature, helping agents respond appropriately to the local design space.

3.3.8 Satisficing behavior

Decision makers tend to satisfice, meaning that they only search the solution space broadly until a solution that satisfies relevant targets is found [70]. Further, engineers and designers tend to have access to such goals [71,72]. Therefore, satisficing may play a crucial role in the design process. The effect of satisficing is implemented in the CISAT framework by increasing an agent's temperature if their designs are far from satisfying relevant targets. The effect of this increase is that the temperature decreases rapidly once a satisficing solution is found, making search more deterministic. However, the temperature remains high until such a solution is found, promoting broad search for a fruitful region of the design space.

3.4 Comparison of CISAT to Human Teams

As a means of validation, the CISAT modeling framework will be used to model the results of the cognitive study from Chapter 2. A review of the original cognitive study is provided below, followed by a description of how CISAT was configured to model the study. The original results of the cognitive study will also be directly compared to the results from the CISAT simulations.

3.4.1 Summary of Human Study

The cognitive study tasked 16 teams of 3 engineering students with the design of a truss structure. Over the course of the study the design problem was changed twice via the introduction of modified problem statements. Participants were given access to a graphical truss design program that allowed them to create, evaluate, and share truss designs within their teams.

Teams designed over the course of six 4-minutes design sessions. The first problem statement (PS1) provided to teams required them to design a truss structure with a factor of safety of 1.25, and a mass as low as possible. Figure 3.2 depicts the two loading points and three supports that were required for every design.

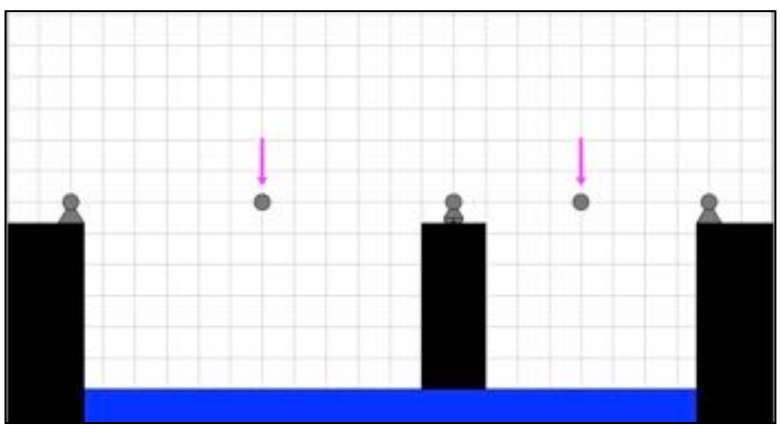


FIGURE 3.2. DIAGRAM OF PROBLEM STATEMENTS 1 AND 2 FOR TRUSS DESIGN STUDY FROM CHAPTER 2.

After working on PS1 for three of the 4-minute design sessions, the participants were given the first of two modified problem statements. PS2 required participants to consider the removal of any one of the bridge supports (leaving only two supports intact at a time). This problem statement required a factor of safety of 1.0, and mass as low as possible.

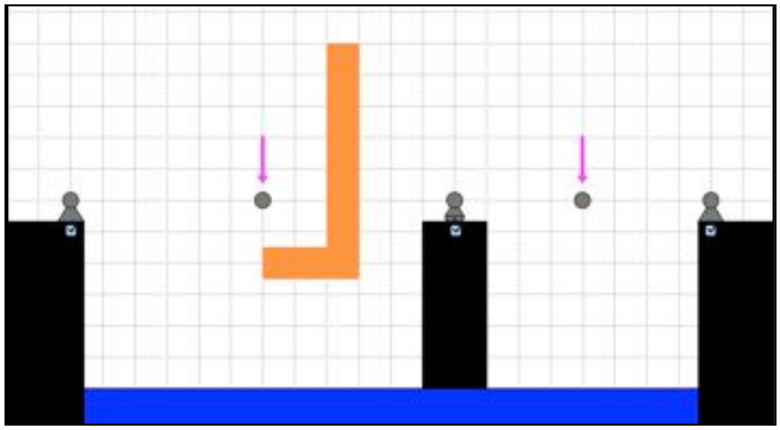


FIGURE 3.3. DIAGRAM OF PROBLEM STATEMENT 3 FOR TRUSS DESIGN STUDY FROM CHAPTER 2.

After working on PS2 for one 4-minute design session, the participants were given the second modified problem statement (PS3). This modification required participants to design their truss around the obstacle shown in Figure 3.3. PS3 required a factor of safety of 1.25, and mass as low as possible. Teams worked on PS3 for the last two 4-minute design sessions.

Participants were also given a mass target for each of the problem statements, thus invoking satisficing tendencies. This mass target will be incorporated into CISAT's satisficing temperature component. Each problem statement also had a constraint on the factor of safety, which will be addressed in the CISAT objective function as a penalty.

3.4.2 Analysis Metrics

Analysis of results from the both the cognitive study and the CISAT simulation will be performed using three metrics. The first metric, *strength-to-weight ratio (SWR) of best design*, tracks the best design produced by a team over time, and was previously defined in Chapter 2. The second metric, *average pairwise distance*, is a means of quantifying divergence (or disagreement) within a team. It is computed as the distance between the designs being explored by any two members in a team at a given instant, averaged across all combinations of two team members. This metric was also defined previously in Chapter 2.

The third metric, *frequency of topology operations*, provides a way to measure how teams change their solution strategies over time. Specifically, it tracks the proportion of topology operations in each 4-minute design session. A topology operation is any operation that modifies the connectivity of the truss (adding or removing joints or members). All other operations are shape operations (changing the size of members or moving joints).

The analysis further involves comparing high- and low-performing teams. Teams are assigned to these two groups based on their cumulative performance across problem statements. The method used to rank teams is identical to that employed in Chapter 2. The top 31.25% of teams are designated as *high-performing teams*, while the lowest 31.25% are designated as *low-performing teams*. The 31.25% cut-off was chosen to match up directly with the cut-off used in the original analysis of the study.

3.4.3 CISAT Configuration

Configuring CISAT to simulate team performance on a problem involves defining appropriate objective functions, implementing a method for instantiating a design, creating move operators to modify designs, and selecting values for other miscellaneous parameters required by CISAT. Simulating this cognitive study requires three objective functions, corresponding to the three problem statements. Every objective function is in the form of the mass of the design plus a penalty to resolve constraints on the solution. The first objective function only places a constraint on the factor of safety (FOS):

$$f^{PS1}(x) = m(x) + g^{FOS}(x). \tag{3.4}$$

In this equation, $m(x)$ indicates the mass of design x, and the function $g^{\mathrm{FOS}}(x)$ computes an appropriate penalty if the factor of safety is too low. The penalty is computed as:

$$g^{FOS}(x) = 10^4 \cdot max\left(0, FOS^{TARGET} - FOS(x)\right)^2. \tag{3.5}$$

FOS_{TARGET} is the factor of safety required for the current problem statement, and $FOS(x)$ computes the factor of safety of truss design x.

The objective function used for the second problem statement, $f_{PS2}(x)$, applies the maximum FOS penalty across a variety of support cases, per PS2:

$$f^{PS2}(x) = m(x) + max\begin{pmatrix} g^{FOS}(x) \\ g^{FOS}(x^{-S1}) \\ g^{FOS}(x^{-S2}) \\ g^{FOS}(x^{-S3}) \end{pmatrix}. \tag{3.6}$$

The notation $x^{-\mathrm{Si}}$ represents the design x with the i^{th} support removed. Therefore, function $f_{PS2}(x)$ applies the highest penalty from all support conditions required in PS2.

The third and final objective function is similar to the initial objective function, but incorporates a second penalty function, $g^{\mathrm{OBS}}(x)$, which penalizes the solution for violating the obstacle defined for PS3 (see Figure 3.3):

$$f^{PS3}(x) = m(x) + g^{FOS}(x) + g^{OBS}(x). \tag{3.7}$$

The second penalty function, $g^{\mathrm{OBS}}(x)$, imposes a penalty based on J_{OBS}, the number of truss joints within the obstacle, and L_{OBS}, the cumulative member length within the obstacle:

$$g^{OBS}(x) = 10^4 \cdot (J_x + L_x^2). \tag{3.8}$$

Agents instantiate their truss designs by first determining how many joints the truss design will have by drawing a random integer. For this work the number of joints is restricted to fall between 8 and 30, inclusive. The location of each joint is chosen so that every joint is approximately equidistant from its nearest neighbors. Delaunay triangulation is then used to determine a stable pattern for connecting the joints using structural members. Only one joint or one member is added per iteration until the initial layout is completed. If an agent's design becomes statically indeterminate over the course of the simulation, it is permitted to instantiate a new truss design.

Move operators must be defined to allow agents to act upon and modify their solutions. The operators defined for this task are listed and described below:

MO1. *Add a member:* The two nearest unconnected joints are connected with a member.

MO2. *Change the size of a member:* A member is selected with probability proportional to $|FOS - FOS_{\mathrm{TARGET}}|$. If at least one member is failing, only failing members are selected from. Once a member is selected, its size is increased if $FOS < FOS_{\mathrm{REQ}}$, and decreased otherwise.

MO3. *Change the size of all members:* If the majority of members have factors of safety greater than FOS_{REQ}, the size of every members is increased. Otherwise, the size of all members is decreased.

MO4. *Delete a member:* A member is probabilistically selected with probability proportional to its factor of safety. The selected member is then deleted.

MO5. *Move a joint:* A joint is selected at random. A greedy and deterministic search algorithm is then used to improve the location of the joint.

MO6. *Delete a joint:* A joint is probabilistically selected with probability proportional to the sum of the factors of safety of all members connecting to the joint. The selected joint is then deleted.

MO7. *Brace a member:* A member is selected for bracing from the set of members that are both in compression, and have a factor of safety less than FOS_{REQ}. A joint is inserted in the middle of the selected member. The new joint is then connected to the nearest joint that it is not already connected to. This move operator is completed over the course of 2 iterations.

MO8. *Add a joint and attach:* A joint location is selected using a procedure identical to that used in MO1. A joint is created at that location, and members are connected between this joint and the three nearest joints. This move operator is completed over the span of 4 iterations.

This set of move operators is not designed to constitute a rigorous design grammar, such as those developed for computational synthesis of frame and truss structures [92,93]. Rather, this set of move operators is designed to reflect the operations available to human participants through the design GUI during the cognitive study (MO1-6), and also to allow simple heuristic combinations of operations (MO7-8).

Participants in the cognitive study had to begin designing their truss by laying out a network of joints and members. Therefore, study participants must have begun with a higher probability of applying move operators that would enable this layout process. To model this aspect in CISAT, the probability of MO1 (adding a member) is increased according to the number of members in the initial layout. Similarly, the probability of MO8 (add and attach a joint) is increased according to the number of joints in the initial layout.

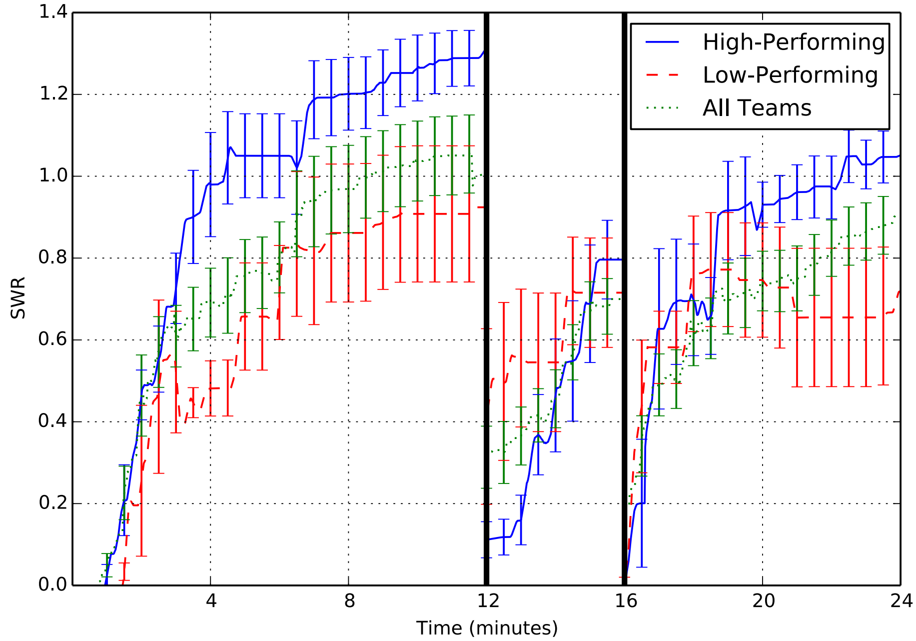
In order to ensure further parity between CISAT simulations and the original cognitive study, agents in CISAT were made to apply move operators at the same average rate as human study participants. An analysis of the data recorded in the original cognitive study indicated that the average individual applied one operation every three seconds. Therefore, over the course of six 4-minute design sessions (1440 seconds), the average individual applied 480 moves. Every CISAT agent was also allowed this number of moves.

3.4.4 Comparison of Results

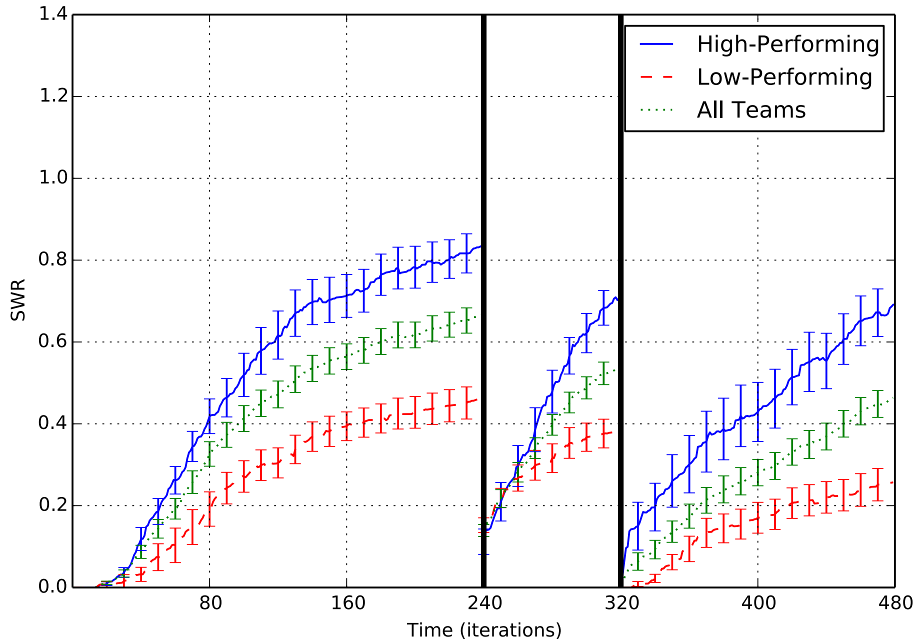
Of the 16 teams that took part in the original cognitive study, 5 were designated as high-performing, and 5 as low-performing, based on an evaluation of their final design solutions. In order to establish a better statistical representation, 64 teams (4 times the number of teams in the cognitive study) are simulated using CISAT. Of these teams, 20 are designated as high-performing and 20 as low-performing. A comparison between the results of the original cognitive study and those simulated using CISAT is provided in Figures 3.4 - 3.6. The vertical, black lines indicate the introduction of a new problem statement.

The CISAT framework reproduces several of the main trends that are apparent in the original results from the cognitive study. For instance, the high-performing human teams showed an early divergent period, followed by a pattern of fairly constant average pairwise distance (see Figure 3.5.c). This is echoed in the CISAT simulation (see Figure 3.5.d). The low-performing human teams show higher average pairwise distance, and a period of divergence near the end of the study. This behavior is also evidenced in the CISAT simulation.

CISAT also predicts the correct mean trend for frequency of topology operations (see Figures 3.6.e and 3.6.f). Both human teams and CISAT teams show an initial decrease in topology operations, followed by an increase after the introduction of the new problem statements. A similar probabilistic transitional pattern from topology operations to parameter operations has been shown as an effective strategy for the computational design of structural frames [94].

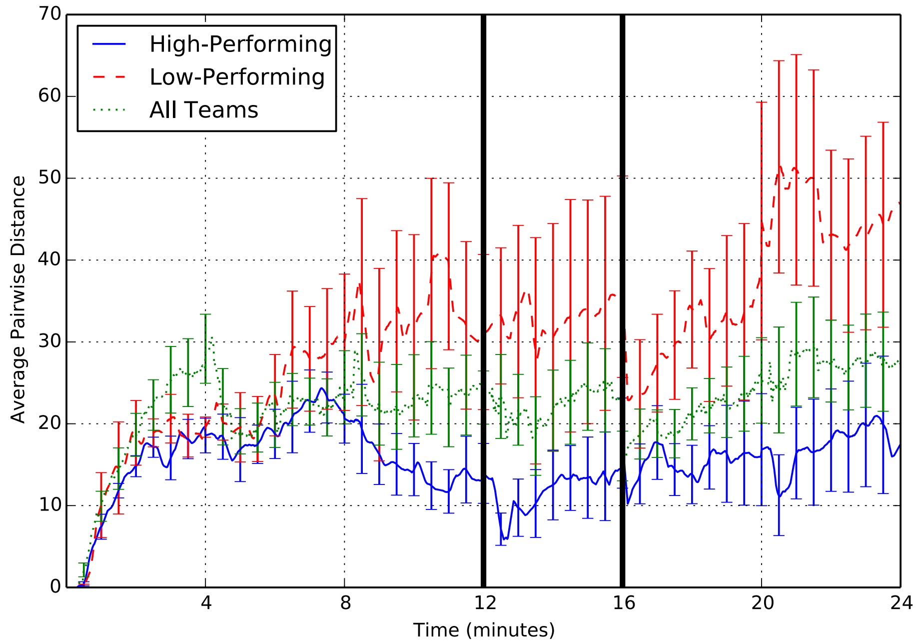


A

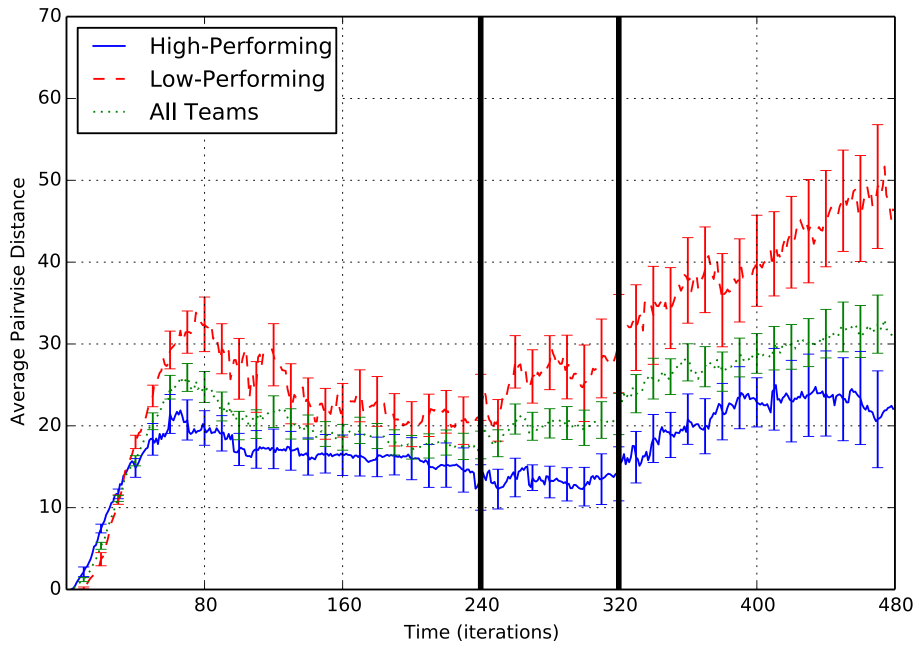


B

FIGURE 3.4. COMPARISON BETWEEN STRENGTH-TO-WEIGHT RATIO OF BEST DESIGN FOR (A) COGNITIVE RESULS FROM TRUSS DESIGN STUDY AND (B) CISAT SIMULATION RESULTS. ERROR BARS SHOW ± 1 S.E.

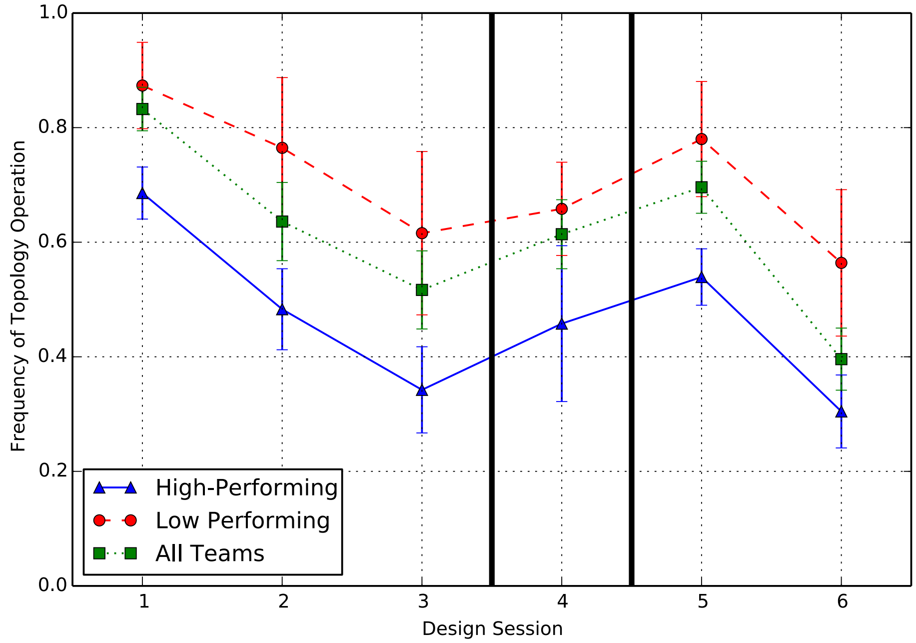


A

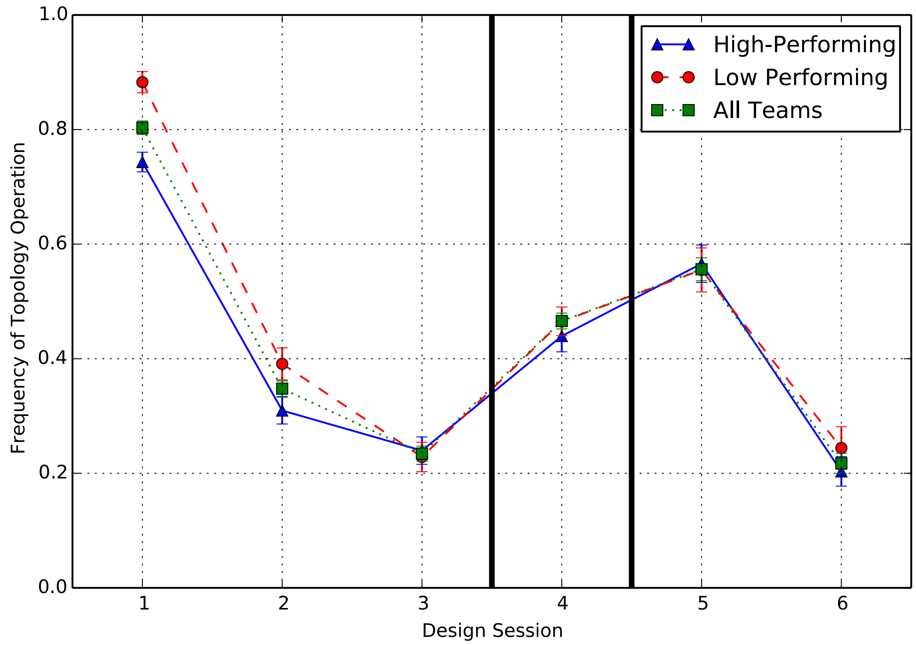


B

FIGURE 3.5. COMPARISON OF DIVERGENCE WITHIN TEAM FOR (A) COGNITIVE RESULS FROM TRUSS DESIGN STUDY AND (B) CISAT SIMULATION RESULTS. ERROR BARS SHOW ± 1 S.E.



A



B

FIGURE 3.6. COMPARISON OF FREQUENCY OF TOPOLOGY OPERATIONS FOR (A) COGNITIVE RESULS FROM TRUSS DESIGN STUDY AND (B) CISAT SIMULATION RESULTS. ERROR BARS SHOW ± 1 S.E.

The Pearson correlation coefficient is used to quantify the degree to which the CISAT framework reproduces the mean trends of the cognitive study. The Pearson correlation coefficient measures the linear correlation between two variables, and returns a value between -1.0 (indicating a perfect negative correlation) and +1.0 (indicating a perfect positive correlation). A summary of Pearson correlation coefficients for the three metrics used for comparison between human and CISAT results is provided in Table 3.1.

TABLE 3.1. SUMMARY OF PEARSON CORRELATION COEFFICIENTS FOR HUMAN VERSUS MODEL RESULTS.

Metric	High-Performing Teams	Low-Performing Teams	All Teams
SWR of Best Design	0.879 ($\rho < 10^{-5}$)	0.786 ($\rho < 10^{-5}$)	0.876 ($\rho < 10^{-5}$)
Average Pairwise Distance	0.676 ($\rho < 10^{-5}$)	0.741 ($\rho < 10^{-5}$)	0.787 ($\rho < 10^{-5}$)
Frequency of Topology Operations	0.951 ($\rho < 0.005$)	0.890 ($\rho < 0.05$)	0.936 ($\rho < 0.01$)

The coefficients provided in Table 3.1 are always above 0.65, and the majority of them are above 0.85. This quantitatively reiterates a fact that is already qualitatively evident in Figures 3.4 - 3.6. Although CISAT does not perfectly reproduce the results of the cognitive study, there is a strong positive correlation between the trends displayed in the two sets of results.

The starkest difference between the CISAT simulation and the original cognitive data is found in the SWR of the best design (Figures 3.4.a and 3.4.b). It is possible that this resulted from the inability of CISAT agents to consider chains of moves. The ability of human problem-solvers to think in terms of multiple sequential moves is indicative of expert problem-solving

[95], but has also been observed in individuals with little experience [56]. Therefore, although CISAT agents were applying move operators in proportions similar to those of the human truss designers, the naïve sequencing of the move operators may have had a handicapping effect. This insight indicates that implementing better models of learning and heuristic development in CISAT could lead to higher performance, and better agreement with human solvers. This hypothesis is addressed in Chapter 7 of this dissertation.

3.5 Investigating Team Strengths with CISAT

In traditional human studies, it is difficult to cull which aspects of problem solving are most influential and beneficial without running multiple studies. Even if multiple studies are possible, it is usually not feasible to isolate features entirely. However, in CISAT such assessment is straightforward and informative. This section focuses on determining the characteristics that were most helpful or harmful in performing the truss design task. By assessing the final SWR of simulated teams composed of agents with and without a given characteristic, it is possible to evaluate the effect of that characteristic on overall performance. For instance, if the removal of a characteristic decreases the final SWR, it can be inferred that the presence of that characteristic is beneficial to team performance. The results of this CISAT analysis then indicate the characteristics that are most important to effective human design teams for, in this case, the truss design problem. This process is shown conceptually in Figure 3.7.

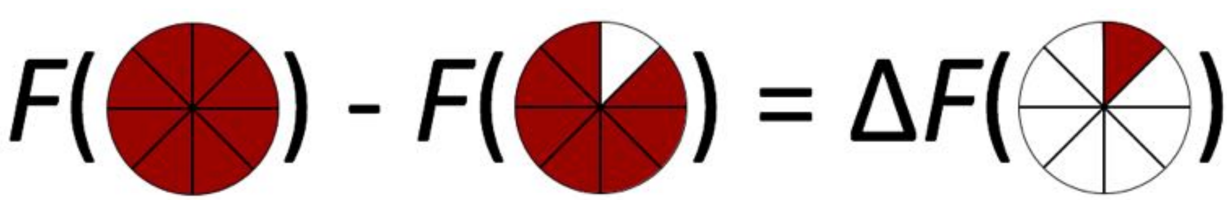


FIGURE 3.7. CONCEPTUAL DIAGRAM FOR COMPUTATION OF CONTRIBUTION FROM INDIVIDUAL CISAT CHARACTERISTICS.

The above procedure is applied to quality bias reduction, operational learning, satisficing, locally sensitive search, self-bias, and organic interaction. The effect of each characteristic is evaluated with 100 simulated teams. In the interest of simplicity, these simulated teams are only used to solve PS1. Multi-agency and solution sharing are not evaluated, since they enable basic team-like performance within CISAT. Figure 3.8 shows the estimated effect on the final SWR from each of the characteristics, relative to the median final SWR of the unmodified CISAT model. The median of the data is used to communicate the results because it is more representative of central tendency than the mean.

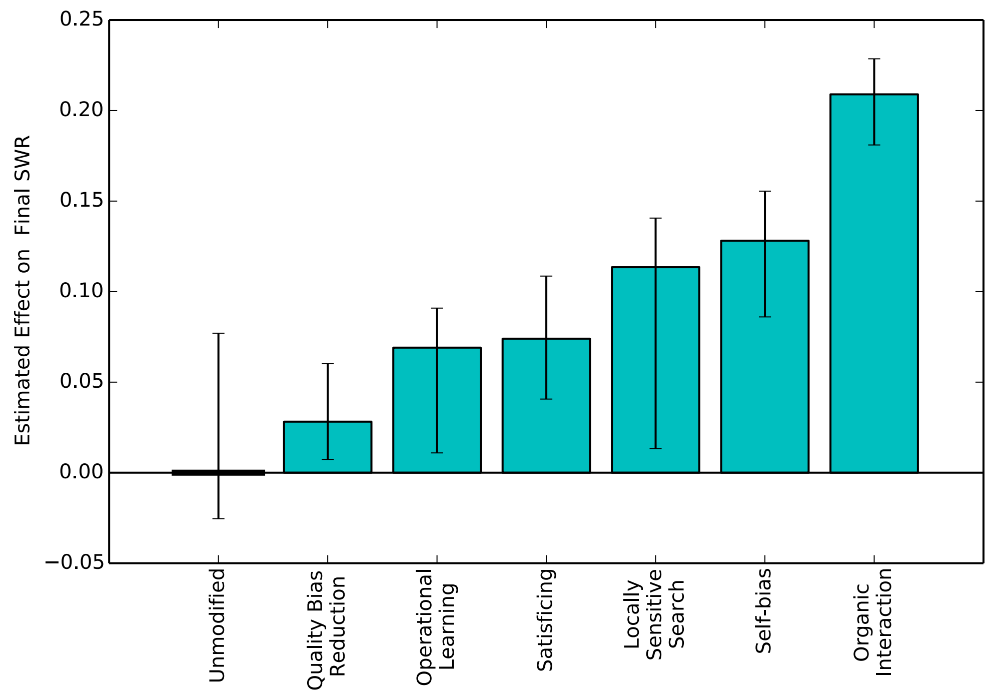


FIGURE 3.8. ESTIMATED EFFECT ON FINAL STRENGTH-TO-WEIGHT RATIO FOR EACH CISAT CHARACTERISTIC. ERROR BARS SHOW ± 1 S.E.

Figure 3.8 communicates the magnitude of median differences, but to further understand the impact of these differences it is necessary to report an *effect size*. Fritz, Morris, and Richler [96] provide a more detailed description of the need for and use of effect size metrics. Although

Cohen's d is commonly used for this purpose, Cliff's δ will be used here because of non-normality in the data [97,98]. Cliff's δ is a non-parametric dominance statistic that indicates the degree to which values in one set of data lie above values in a second set [99]. A value of -1 or +1 indicates no overlap between sets (complete dominance), while a value of 0 indicates complete overlap (no dominance). The effect sizes for the comparative analysis performed in this chapter are provided in Figure 3.9.

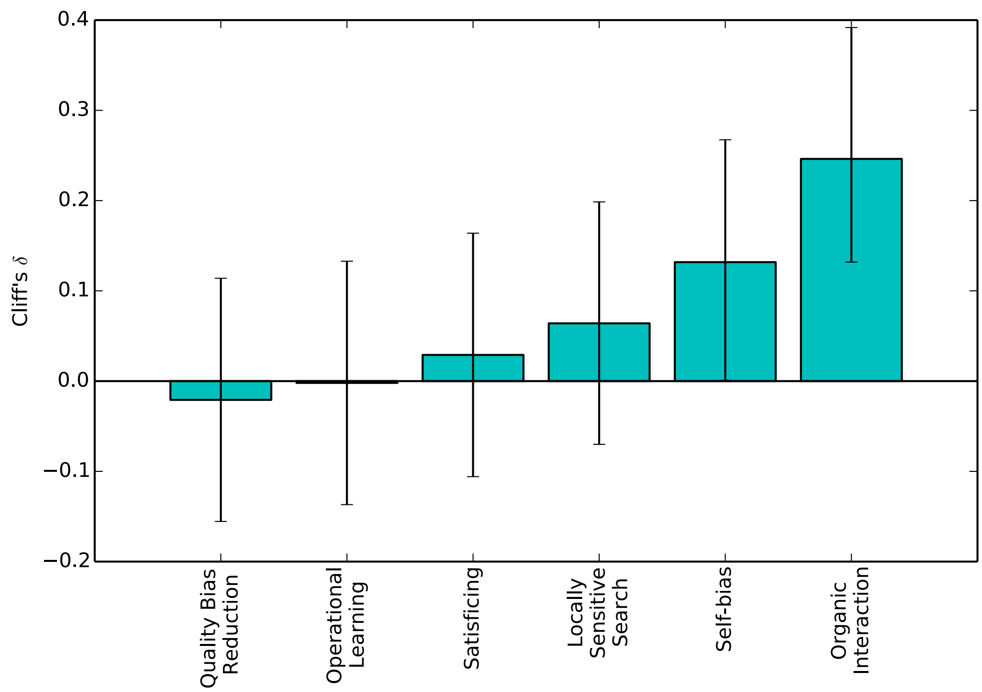


FIGURE 3.9. CLIFF'S DELTA EFFECT SIZE ON FINAL STRENGTH-TO-WEIGTH RATIO FOR EACH CISAT CHARACTERISTIC. ERROR BARS SHOW 90% CONFIDENCE INTERVALS.

As discussed in previous sections of this chapter, each of the characteristics analyzed in this section has been observed in humans. Figures 3.8 and 3.9 show that self-bias and organic interaction display the largest and most significant effects. Both of these characteristics specifically play a role in moderating interaction between team members or agents. These two characteristics decrease the frequency and effect of interaction when present. The absence of organic interaction increases the frequency of interaction (returning the agent team to a default high-interaction state), and the absence of self-bias increases the likelihood that an agent will abandon its current solution in favor of a solution being pursued by a teammate.

Examining the performance attributes of teams without these beneficial characteristics could offer further insight as to the cause of their poor performance. Figures 3.10 and 3.11 show the median values of the SWR of the best design and average pairwise distance during solving.

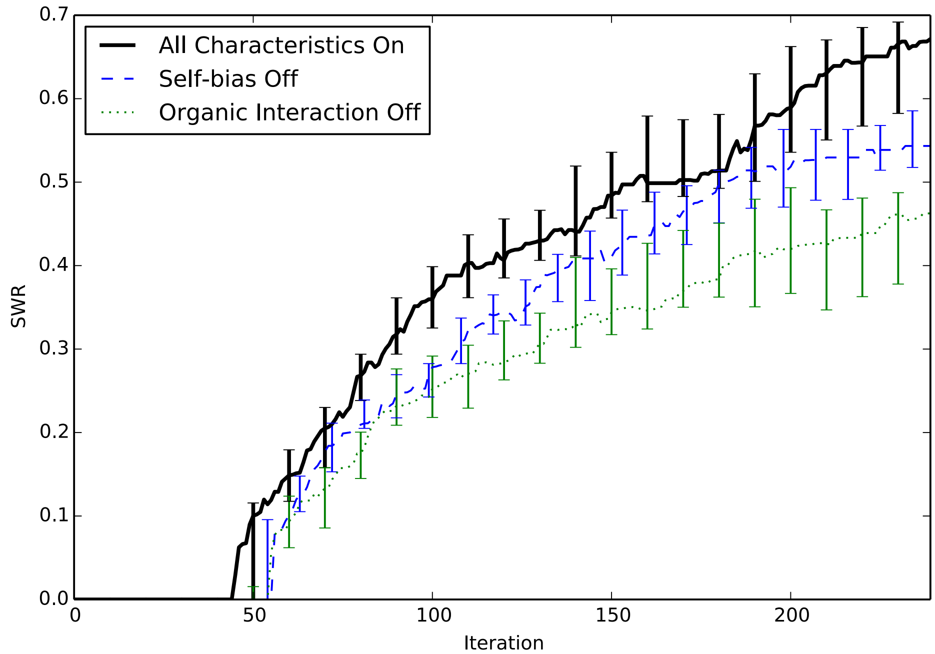


FIGURE 3.10. COMPARISON OF STRENGTH-TO-WEIGHT RATIO OF BEST DESIGN SHOWING EFFECT OF MOST IMPACTFUL CISAT CHARACTERISTICS. ERROR BARS SHOW ± 1 S.E.

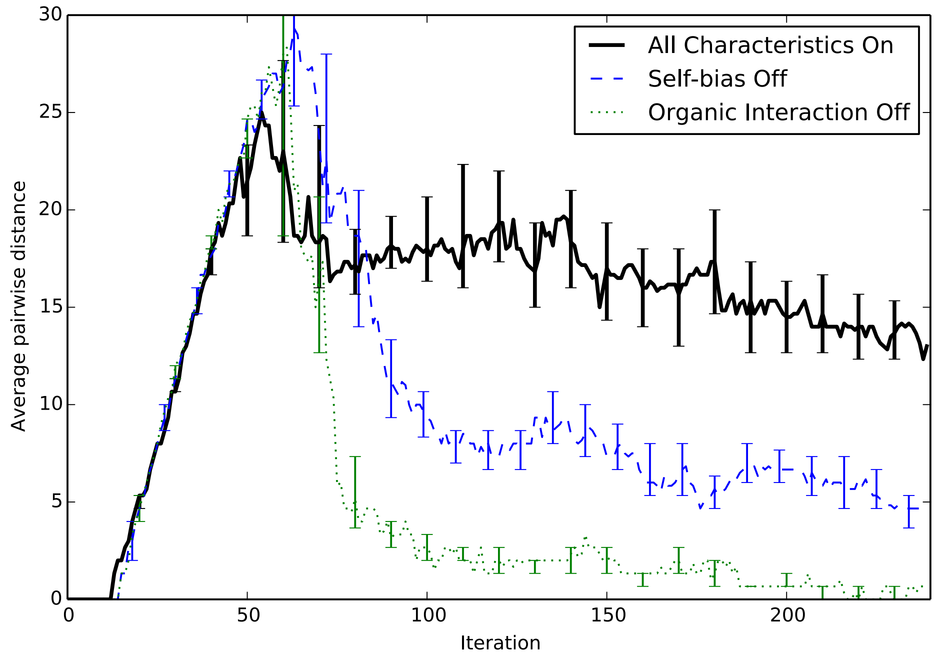


FIGURE 3.11. AVERAGE PAIRWISE DISTANCE SHOWING EFFECT OF MOST IMPACTFUL CISAT CHARACTERISTICS. ERROR BARS SHOW ± 1 S.E.

Figure 3.11 indicates that typical teams (teams simulated with all characteristics turned on) achieve high performance through a period of slow convergence. In contrast, the poorly performing teams converge quickly, resulting in lower final solution quality. This indicates that both frequent interaction and low self-bias may lead to premature convergence within teams, precipitating final design solutions with lower quality. This may also imply that exploring design methods that encourage divergence may improve final design solution quality by further protecting teams from premature convergence.

Additional statistical values describing the final SWR are shown in Table 3.2. These include the 25th percentile (or first quartile point), the 50th percentile (or median) and the 75th percentile (or third quartile point). The inter-quartile range (difference between the 75th percentile and 25th percentile) is shown to indicate the spread of the data within a group of teams, and the inter-group range is shown to indicate the range of values displayed across groups of teams. Examining these statistics can indicate the way in which the teams' final SWRs are distributed about the median. The 75th percentile is simply the median of the upper half of the data, so it is indicative of the performance of the best teams. Similarly, the 25th percentile is indicative of the performance of the worst teams. In Table 3.2, the value of the 25th percentile displays little difference between groups (an inter-group range of approximately 0.054). However, there is much greater variability between the upper quartiles of the groups (demonstrated by an inter-group range of 0.144). In addition, the inter-quartile range is much higher for the group of teams in which all characteristics are turned on. This indicates that removing characteristics from the CISAT simulation framework does not substantially impact the low-performing teams, but does handicap the performance of the high-performing teams. This consequentially compresses the distribution, as evidenced by the decrease in inter-quartile range.

TABLE 3.2. DESCRIPTIVE STATISTICS FOR FINAL STRENGTH-TO-WEIGHT RATIO FOR MOST IMPACTFUL CISAT CHARACTERISTICS.

Statistic	All Characteristics On	Self-bias Off	Organic Interaction Off	Inter-group Range
25th percentile	0.400	0.398	0.346	0.054
Median	0.672	0.543	0.462	0.210
75th percentile	0.854	0.779	0.710	0.144
Inter-quartile Range	0.454	0.381	0.364	-

Because self-bias and organic interaction play a role in moderating communication between agents, these results emphasize the crucial role of interaction in human teams [50,74,75]. Self-bias encourages individuals to continue work on their current solution concept, adding significant detail and critical refinement. Organic interaction decreases the frequency of communication, therefore increasing the extent to which individuals refine their current solutions between interactions.

The benefit derived from the presence of organic interaction and self-bias can be understood by appealing to concepts used in the analysis of social networks. A social network is composed of individuals and the relationships, or ties, between them. These ties can be characterized in terms of strength [100]. Strong ties exhibit frequent interaction, reciprocity, and high emotional intensity, while weak ties have infrequent interaction, little reciprocity and lower intensity [100]. There is evidence that weak ties may play a crucial role by transmitting information between individuals with diverse perspectives and diverse approaches to problems [101]. In the context of the truss study, the presence of organic interaction and self-bias decrease the frequency and intensity of interactions, thus forming weak ties. In turn, these weak ties likely allow team members to explore different regions of the design space, leading to higher divergence and final design solutions with higher quality.

Conversely, the absence of either organic interaction or self-bias can increase the strength or frequency of interactions, leading to the formation of strong ties. It has been demonstrated that strong ties promote group cohesion, but may do so at the expensive of the group's goals [102,103]. Within the context of the truss design problem, this could lead to a state in which the formation of consensus (or low divergence) between team members becomes more of a driving factor in the design process than the search for high quality solutions. Such a state is similar in many ways to groupthink, a psychological phenomena characterized by the search for consensus with little regard for critical evaluation of concepts [17]. It has been theorized that groupthink may detrimentally affect decision-making teams [18].

Up to this point in the chapter, analyses have made comparisons between several groups of teams, where all agents within a group of teams had the same active characteristics (either all characteristics on, self-bias turned off, or organic interaction turned off). Now, trends *within* groups of teams will be examined by computing the correlation coefficient between final average pairwise distance and final SWR (both of which are team characteristics). For teams simulated with all agent characteristics turned on, the Pearson correlation coefficient between these two variables is -0.446. For teams in which agents lacked self-bias, the correlation coefficient is -0.481, and for teams in which agents lacked organic interaction it is -0.301. Similar negative correlations are observed with the removal of other characteristics, and all correlations are highly significant ($\rho < 0.005$). This indicates that low final average pairwise distance tends to occur in teams that also have a high final SWR. Therefore, relative to other teams in which agents have the same active characteristics, a team that shows low final divergence is likely to produce a high quality solution.

A similar trend has been demonstrated in a variety of design problems. For instance, team mental model sharedness was correlated with coordinated team performance in teams playing the open-ended Kantjil Design Game [104]. The correlation was also identified when the truss design problem from this chapter was solved by human designer (see Chapter 2). There, the relationship was attributed to expert-like characteristics of high-performing teams. Expert designers have been shown to quickly commit to a single solution concept [42]. The fact that expert solutions also tend to be of high quality is indicative of the fact that expert designers are capable of selecting a good initial representation of the problem, and do not need to search divergently. Although the agents that compose the CISAT modeling framework are not intended to be experts, they are still created with a variety of initial representations with varying levels of quality. A team with a lower quality representation may need to search divergently in an attempt to improve that representation. However, a team with a high quality representation has no need to refine their representation through broad search. Thus, the same mechanism (selection of initial representations with varying levels of quality) may have given rise to a similar trend (negative correlation between divergence and quality) in both human- and agent-based studies.

3.6 Summary

This chapter introduced the Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework, an agent-based platform for simulating team-based engineering design. This framework was used to directly simulate the results of a cognitive study in which teams of engineering students tackled a structural design problem. A comparison of the CISAT simulated results to those of the original cognitive study from Chapter 2 revealed a high degree of linear correlation between the two. This indicates that CISAT is capable of capturing the trends observed in humans solving a simplified engineering design task. Next, CISAT was used to explore the particular characteristics that were most beneficial to teams in solving this task. This analysis indicated that proper interaction (specifically self-bias and organic interaction timing) was crucial to enabling team success in the truss design task. Further analysis revealed the importance of flexible design methods that allow for sufficient, but not excessive, divergent search.

This current implementation of CISAT only models a set of cognitive phenomena that have been demonstrated within the domains of design or problem-solving. However, the validated CISAT framework can now be used as a platform to simulate the effects of phenomena that have not been explicitly demonstrated in those domains. The results of such simulations could be used to formulate promising studies to be carried out with human test subjects.

The current validation study has demonstrated that the CISAT modeling framework is capable of accurately modeling small teams. However, larger teams may require a hierarchical structure amongst members, particularly if solving a complex task. Although CISAT is currently only capable of simulating flat teams (i.e. teams without overt structure), it may act as a building block for simulating larger, hierarchical organizations. For instance, an organization composed of multiple sub-teams could be modeled as a conglomeration of CISAT teams. Additional development would be necessary to define an inter-team protocol for communicating solution information, and to define specific responsibilities and sub-tasks for individual teams. In this context, biased information passing [105] between teams may emerge as an important inter-team characteristic.

Subsequent chapters of this dissertation will apply the CISAT modeling framework to address several areas of design research. These chapters specifically use CISAT to examine quantifiable methods for selecting optimal characteristics for human teams (Chapters 4 and 5) to create a team-inspired optimization algorithm (Chapter 6), and to study the effects of sequence learning capabilities on designers (Chapter 7).

Chapter 4: Optimizing Design Teams: Theory Development via Computational Modeling[3]

We adore chaos because we love to produce order.

M.C. Escher

4.1 Overview

It is commonly assumed that interacting teams are more effective than non-interacting groups of individuals. However, teams with the wrong characteristics may in fact function much *less* effectively than the sum of individuals in certain situations [17,19]. It is therefore crucial for a team to have the right characteristics in order to achieve maximum effectiveness. This chapter uses the CISAT computational framework to investigate how the properties of a design problem can be used to inform the selection of the best team characteristics for solving it. The CISAT framework is used in place of traditional cognitive studies because the development of this relationship requires team performance to be assessed on a large variety of different design problems, and with respect to a variety of different values for a set of team characteristics. This quickly compounds the number of conditions that must be evaluated, resulting in a research study requiring an unmanageable number of participant-hours.

---

[3] A portion of this chapter is based on:

McComb, C., Cagan, J., and Kotovsky, K., 2016, "Linking Properties of design problems to optimal team characteristics," ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Charlotte, NC, USA.

The primary objective of this chapter is to establish a set of equations that relate the properties of a given design problem to the best team characteristics for solving that problem. Three cases are specifically addressed. Case 1 considers the scenario in which a manager must determine the best team size as well as the best interaction frequency for that team. This scenario most likely occurs in larger design firms that permit the flexibility necessary to assemble ad hoc teams. Case 2 addresses the scenario in which a design team with a fixed size must determine how frequently to interact while solving the problem at hand. This is a more likely scenario in smaller firms that have less flexibility. Case 3 examines the scenario in which the team will interact at a fixed frequency, and the best team size must be chosen. This case might apply if the members of a team are not co-located and meetings are scheduled intermittently, resulting in a fixed interaction pattern. Several stages of work are necessary to produce predictive relationships for these three cases:

1. A set of design problems is first defined (see Section 4.3).
2. The properties of these design problems are then computed (see Section 4.4).
3. The CISAT framework is used to find the best team characteristics for solving each problem (see Section 4.5).
4. Regression analysis is used to define equations that allow optimal team characteristics to be predicted based on problem properties (see Section 4.6).

The end result of this process is a tool that can theoretically be used to inform the selection of team size and interaction frequency for solving a given design problem. Although team size and interaction frequency are the focus of this chapter, the method can be used to examine and design other team characteristics that can be tested within the CISAT framework.

4.2 Background

A variety of definitions for the word *team* have been supplied in the literature [6–9], but two concepts are pervasive across these definitions: multi-agency (the composition of a team from two of more individuals) and communication (the ability of those individuals to exchange information). These two concepts are central to the nature of teamwork, and investigating their relationship to team effectiveness should provide fundamental insights into the team-based search for solutions. This work operationalizes the concepts of multi-agency and communication by specifically investigating the impact of ***team size*** and ***frequency of interaction***. These two characteristics are of specific interest for engineering design because they help to highlight the tension that exists between breadth and depth of search in the design space. Larger teams can enhance the breadth of search for solutions, but extremely divergent search may be unnecessary and wasteful for some problems [36]. More frequent interaction leads to deeper and more focused search, but it can also lead to design fixation [37]. The appropriate selection of values for these characteristics ensures that teams are able to diverge and converge in a way that is appropriate for the task at hand. The results of this research provide a means for selecting appropriate values for these characteristics before work begins on a design problem.

Team size plays a role in the search for solutions, but prior findings are mixed. Studies that report negative results for larger teams typically find that larger teams are plagued by low efficiency and coordination issues [106–108]. In contrast, other work has shown that larger teams may benefit from concurrent team work and a greater breadth of experience and opinions [109–111]. A meta-analysis of team characteristics showed that there is a small positive relationship between team size and performance [112]. However, a more detailed analysis indicated that the relationship between team size and performance depends on the type of team –

teams brought together to accomplish a finite project benefit from larger sizes, but teams that work together continuously do not [112]. It was also discovered that management teams benefit from larger sizes [112]. This aligns with other work indicating that optimal team size may be task- or at least domain-dependent [113,114].

In practice, many design tasks are limited by constraints on human power, (e.g., a fixed number of billable hours available for a project). This raises a perennial question: should resources be concentrated within a small team, spread among many individuals in a larger team, or something in between? With respect to software development, research has shown that the answer to this question depends in part on how easily the project can be partitioned into subtasks, and whether or not significant communication overhead is necessary after partitioning [115,116]. Analysis of completed projects has shown that while increasingly complex projects demand larger teams, larger teams also tend to be less efficient [117]. This implies that there exists an optimal team size that depends on project properties. This possibility led to the creation of a theoretical model relating the optimal size of software development teams to the predicted size of the project [22].

The impact of team size has also been explored in other domains. Computational work in social choice theory has shown that smaller teams are capable of making decisions which more fairly represent the preferences of the team [118]. In addition, work on team-inspired agent-based optimization algorithms has shown that the optimal agent team size depends on the objective function (see Chapter 6).

The ***frequency of interaction*** is a common measure of communication within a team [119–122]. It has been shown that the relationship between frequency of interaction and task performance is approximately quadratic in cross-functional teams [123]. High and low interaction frequencies result in lower performance, with a well-defined optimal interaction frequency [123] lying between. A computational model was developed to investigate this phenomenon further, and it indicated that higher interaction frequency tended to decrease the quality of communication [124]. Further computational work demonstrated a relationship between optimal interaction frequency and project complexity [21].

With respect to design, the benefit derived from interaction generally arises from the ability of individuals in a team to explore a variety of options, but then to collaboratively focus their attention on a shrinking set of the most promising alternatives [44]. However, interaction is not always beneficial when it is allowed. Computational simulations indicate that excessively frequent interaction can be detrimental to teams, resulting in the implicit prioritization of consensus over the search for good solutions (see Chapter 3). This shift in priorities within a team is similar to a psychological phenomenon known as groupthink [17], which can be harmful to decision-making teams [18]. In contrast, less frequent interaction may lead to the formation of weak ties between members of a team (see Chapter 3). Weak ties can be beneficial because they facilitate the transmission of diverse perspectives between individuals [101]. Other research has examined how team interaction can be structured to make teams more resilient to change [74]. Results suggested that encouraging interaction between individuals with diverse opinions could weaken confirmation bias, thus improving performance [74]. These studies indicate the importance of selecting an appropriate level of interaction to encourage independence and exploration while still allowing for the exchange of beneficial ideas.

In a study of the connection between problem formulation and creative ideation outcomes [125], participants were presented individually with a conceptual design problem. Characteristics of their problem formulation process were tracked using the P-map framework [126], and the outcomes of their work were quantified with respect to the ideation effectiveness metrics developed by Shah et al. [127]. Regression analysis was then used to relate the P-map variables to those ideation metrics, providing a predictive relationship [125].

The work presented in this chapter also uses regression to provide predictive equations, but focuses on predicting the team characteristics that lead to the best solutions. Further, this chapter does not involve human studies to determine the relationship, but instead utilizes the CISAT modeling framework. Without CISAT, this research would have been excessively time consuming and prohibitively expensive. The CISAT framework is used here to simulate the performance of 1,120 conditions, with 100 teams in each. On average, each simulated team conducts 1,250 operations during solving, and human participants perform operations at a rate of approximately 1,200 operations per hour (see Chapter 3). Therefore, the simulation results presented in this chapter represent the equivalent of more than 100,000 participant-hours of human studies.

4.3 Design Problem Definitions

The relationship between design problem properties and optimal team characteristics is studied using both fluid network and structural configuration design problems. These problem classes are used because their solutions are dictated by dissimilar physical phenomena, guaranteeing a broad range of problem characteristics. The problems also lend themselves well to computational design since the quality of potential solutions can be readily quantified. Within each problem class, four design problems are defined, with the intent of providing a variety of different problems within the class.

Example solutions for the structural design problems are shown in Figure 4.1. The problem types, all trusses, include two tower-style problems with both vertical and lateral loads

(Figures 4.1.a and 4.1.b), a single-span bridge problem (Figure 4.1.c), and a double-span bridge problem (Figure 4.1.d). Pin supports (which resist both vertical and horizontal translation) are denoted by a solid triangle, while roller supports (which only resist vertical translation) are denoted by a black triangle on top of two circles. Arrows are used to denote loads.

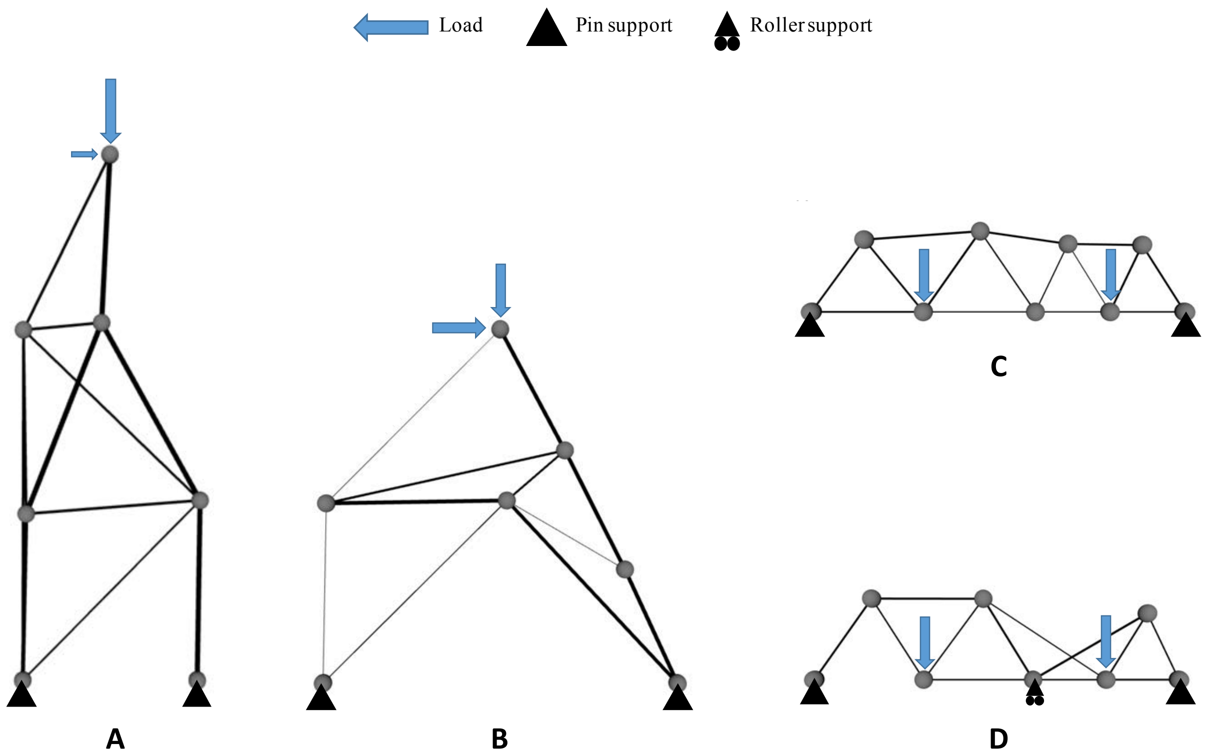


FIGURE 4.1. EXAMPLE SOLUTIONS TO STRUCTURAL DESIGN PROBLEMS, SHOWING REQUIRED LOADS AND SUPPORTS: (A) NARROW-BASE TOWER LAYOUT, (B) WIDE-BASE TOWER LAYOUT, (C) SINGLE-SPAN BRIDGE LAYOUT, (D) DOUBLE-SPAN BRIDGE LAYOUT.

These structural design problems charge CISAT simulated teams with maximizing the factor-of-safety of their solutions while minimizing the mass. Support location and type are specified for each of the problems, and cannot be modified by CISAT agents. The location, magnitude, and direction of loads are similarly specified and immutable. CISAT agents are permitted to act upon solutions by adding and removing joints, adding and removing structural members, changing the size of members, and changing the location of joints (provided that the joints are neither supports nor loaded).

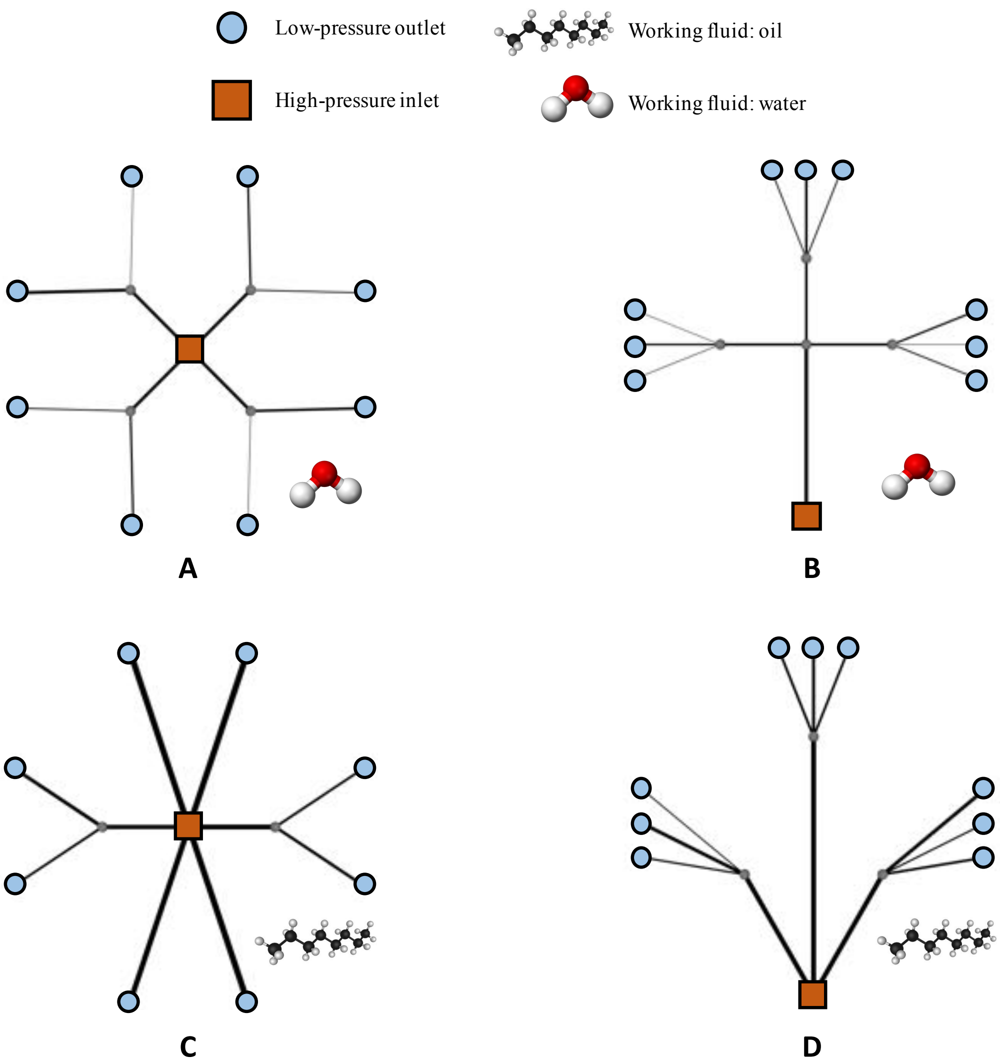


FIGURE 4.2. EXAMPLE SOLUTIONS TO FLUID CHANNEL DESIGN PROBLEMS, SHOWING PRESSURES AT REQUIRED INLETS AND OUTLETS FOR (A) CONCENTRIC WATER DISTRIBUTION NETWORK, (B) ECCENTRIC WATER DISTRIBUTION NETWORK, (C) CONCENTRIC OIL DISTRIBUTION NETWORK, AND (D) ECCENTRIC OIL DISTRIBUTION NETWORK.

Example solutions to the fluid network design problems are shown in Figure 4.2. The arrangement of inlets and outlets is specified as either concentric or eccentric. A concentric layout indicates that the high-pressure inlet is placed near the geometric center of the low-pressure outlets (Figures 4.2.a and 4.2.c). An eccentric layout denotes that the high-pressure inlet is placed far away from the center of the low-pressure outlets (Figures 4.2.b and 4.2.d). The working fluid for the network is also specified as either water (Figures 4.2.a and 4.2.b) or oil (Figures 4.2.c and 4.2.d). These fluids differ greatly in viscosity, impacting the structure of the design space. Laminar flow is assumed in order to afford quick closed-form evaluations of solution quality.

These fluid design problems require CISAT simulated teams to maximize the flow rate at each of the outlets while minimizing the total length of pipe used for the solution. The location and pressure of inlets and outlets are specified as part of the problem statement and cannot be changed by CISAT agents. Agents are permitted to modify solutions by adding or removing junctions, adding or removing lengths of pipe, and changing the diameter of pipes.

To effectively solve these multi-objective problems, CISAT agents were made to select a weighting of the objective functions, chosen randomly from a uniform distribution between 0 and 1. This facilitated the development of unique solution preferences within simulated design teams.

4.4 Characterization of Design Problems

The nature of each design problem is quantified with respect to three properties, each of which provides information that is informative for selecting team characteristics. These properties are:

1. the alignment between the objective functions of the problem (c_A), which increases the importance of search breadth (particularly if objective functions disagree);

2. the local structure of the design space (c_L), which measures local roughness of the design space, and can therefore limit how efficiently an individual is able to search for local minima; and
3. the global structure of the design space (c_G), which measures modality of the space, and thus bears on the extent to which a team has to coordinate their search of the space.

These three properties can be computed by taking a random walk of finite length through valid solutions in the design space. The solutions traversed during the walk are then evaluated with respect to each of the objective functions associated with the current design problem, with the results stored in separate vectors. This result is a set of vectors $\{\boldsymbol{Y}^1, \boldsymbol{Y}^2, \ldots, \boldsymbol{Y}^N\}$, where N is the total number of objective functions. The vector $\boldsymbol{Y}^i$ contains values along the i^{th} objective function for the solutions traversed during the random walk. The $\boldsymbol{Y}^i$ vectors will be used below to offer mathematical definitions of the three properties.

4.4.1 Objective function alignment

Engineering design often necessitates the consideration of multiple objectives for a given design problem [128]. One can imagine a scenario in which a machine part is being designed with the objectives of minimizing mass while simultaneously minimizing cost. If the total cost of a part is driven by the cost of bulk material, then these objectives are aligned – they may even be related by a constant value (the per-mass cost of material). To illustrate the other extreme, imagine a scenario in which the mass of the part may only be decreased by machining away material. In this case, the cost of the part would likely be dominated by machining costs, so the objectives of minimum mass and minimum cost would be opposed. If objective functions disagree considerably, a team may need to search more divergently in order to discover a region of the design space in which objective functions are more aligned or a region in which all objective functions reach acceptable values. Less frequent interaction enables team members to pursue their own solutions in detail, potentially allowing team members to search the space divergently.

Objective function alignment can be quantified as the average pairwise Spearman correlation between the sampled values for each combination of objective functions. The characteristic value describing this quantity is computed as

$$c_A = \underset{\forall i,j:i \neq j}{mean}\, \rho_S\left(\boldsymbol{Y}^i, \boldsymbol{Y}^j\right), \tag{4.1}$$

where ρ_S denotes the Spearman rank correlation between objective functions i and j for the sampled solutions. Spearman's rank correlation coefficient is a non-parametric measure of the correlation between two samples [129]. Rank correlation is used in lieu of linear correlation because an ordinal relationship is sufficient to indicate alignment between objective functions.

A value of $c_A < 0$ indicates that the objective functions show some level of misalignment or opposition. As noted above, this would necessitate a higher level of divergence, which could be facilitated by infrequent interaction. A value of $c_A > 0$ indicates that the objective functions show a meaningful degree of alignment ($c_A \equiv 1$ if there is only one objective function). In this case, a team could benefit from frequent interaction and enabling a quicker, convergent search for the solution.

4.4.2 Local Structure

A fractal is a pattern that exhibits local self-similarity, meaning that similar patterns emerge across scales. Fractal-like patterns have been identified in a number of fields [130,131], and have been noted as a distinguishing characteristic of layout problems [132,133] and may therefore be important in characterizing the structure of design spaces in a more general sense. Computing the fractal dimension, D, of a random walk along a function reveals the local scaling relationship that the function follows. In general, a low fractal dimension indicates a locally smooth curve, while a higher value indicates roughness (see [134] for examples of how function topology changes with fractal dimension). For this work, the local structure property is defined as the maximum fractal dimension observed across objective functions:

$$c_L = \max_i D(\boldsymbol{Y}^i). \tag{4.2}$$

In other words, c_L is the fractal dimension of the roughest objective function. A large value of c_L indicates a design problem with at least one locally rough objective function, while a lower value of c_L indicates that all objective functions are locally smooth (and perhaps traversable with gradient methods). When a design space is locally rough, the local minima are not easy to find since gradient methods cannot be used. Therefore, infrequent interaction within a team could be beneficial, allowing individuals to intensively search different neighborhoods in the design space.

4.4.3 Global Structure

While the fractal dimension can be used to define the local structure of a design space, the Hurst exponent expresses global structure. Together, these two properties provide a robust depiction of design space behavior across scales. Specially the Hurst exponent, H, expresses the long-term memory of a time series [135]. A Hurst exponent near 1 indicates that a high value is likely to be followed by another high value, while an exponent near 0 indicates that a high value is likely to be followed by a low value. Computing the Hurst exponent of a random walk can be indicative of the global roughness of the landscape – a value of $H = 1$ indicates that the time landscape is globally smooth, and $H = 0$ indicates global roughness. This also correlates approximately to the modality of the function – a lower value of H reveals a multimodal landscape (see [134] for examples of how function topology changes with H). For this work, the global structure property is defined as the minimum Hurst exponent observed across objective functions:

$$c_G = \min_i H\left(\boldsymbol{Y}^i\right). \tag{4.3}$$

In other words, c_G is the Hurst exponent of the most multimodal objective function. A value of c_G near zero indicates a design problem with at least one objective function that is highly multimodal. A value of c_G near one indicates that all objective functions have few local optima. As a function becomes more multimodal, the team has to search broadly to find and evaluate local minima. Breadth of search like this could be enhanced by a low level of interaction between team members. This would encourage independent search of the design space, which in turn would delay convergence to a common solution.

4.4.4 Example Characterization

Figure 4.3 shows an example of a random walk taken through a design space with two objective functions. Based on this random walk, the first objective function yields a Hurst exponent of $H_1 = 0.38$ and a fractal dimension of $D_1 = 1.50$, while the second function yields a Hurst exponent of $H_2 = 0.49$ and a fractal dimension of $D_2 = 1.10$.

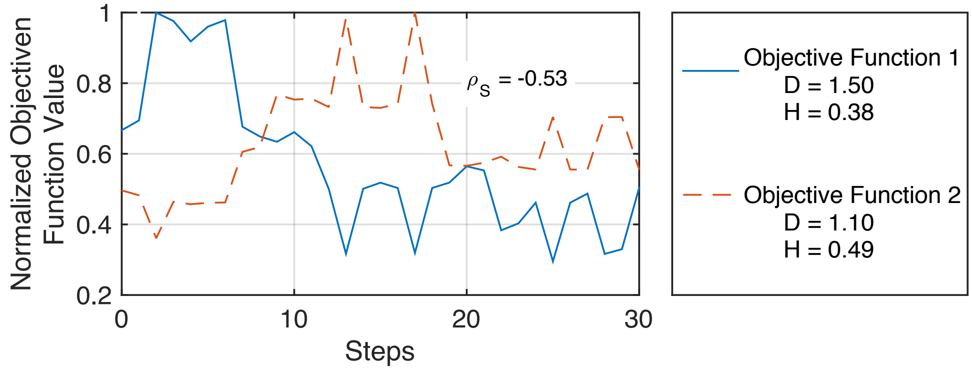


FIGURE 4.3. EXAMPLE RANDOM WALK WITH COMPUTED PROBLEM PROPERTIES.

Based on the above values, the global structure property can be computed as $c_G = \min(H_1, H_2) = 0.38$ and the local structure property can be computed as $c_L = \max(D_1, D_2) = 1.50$. Further, the alignment property can be determined by computing the Spearman correlation coefficients between the two objective functions, $c_A = \rho_S(\boldsymbol{Y}^1, \boldsymbol{Y}^2) = -0.53$.

The properties of each design problem in this work are determined by computing the mean values each of c_A, c_L, and c_G obtained from 100 separate random walks. The repetition of the random walks ensures that the properties are estimated with high accuracy, thus reducing a possible source of error in the subsequent regression analysis.

4.5 Finding Optimal Team Characteristics

This sections details how the optimal team characteristics are found for both Case 1 (in which both team size and interaction frequency must be chosen), Case 2 (in which the team size is fixed, and interaction frequency must be chosen), and Case 3 (in which the interaction frequency is fixed, and team size must be chosen). First, team performance is assessed with the CISAT modeling framework for every combination of design problem (8 problems defined in Section 3), team size T (from 2 to 6), interaction frequency F (values of 0, 1/32, 1/16, 1/8, 1/4, 1/2, and 1, indicating the fraction during which teams interact), and total number of solution evaluations R (values of 500, 100, 1500, and 2000). The variable R is analogous to the number of billable hours available for a design project, and provides a critical limitation on the resolution with which the space can be searched. R will be referred to as resource availability in the remainder of this chapter.

For every combination of the above variables, the CISAT modeling framework is used to simulate 100 design teams. A post-processing step is used to determine the fraction of teams that were able to achieve at least one solution that met the target values for all objective functions. This fraction is the criterion for selecting the best team characteristics. Further post-processing (outlined in the next two paragraphs) is used to extract sets of data for the regression analyses.

Case 1 corresponds to a situation in which both team size and interaction frequency must be chosen. To do so, resource availability and the design problem are held constant, and all combinations of team size and interaction frequency are examined to find the optimal combination. This process is illustrated visually in Figure 4.4.a. Contours show the fraction of teams that were able to meet all design targets. The optimal interaction frequency (F_{OPT}) and the optimal team size (T_{OPT}) are chosen so that they maximize this fraction. When applied across all simulations this procedure results in a data set of 32 observations (8 design problems × 4 values of R). Every observation in this data set consists of values for 2 dependent variables (the team characteristics, T_{OPT} and F_{OPT}) and 4 independent variables (c_A, c_G, c_L, and R). This data set forms the basis for the regression analysis for Case 1.

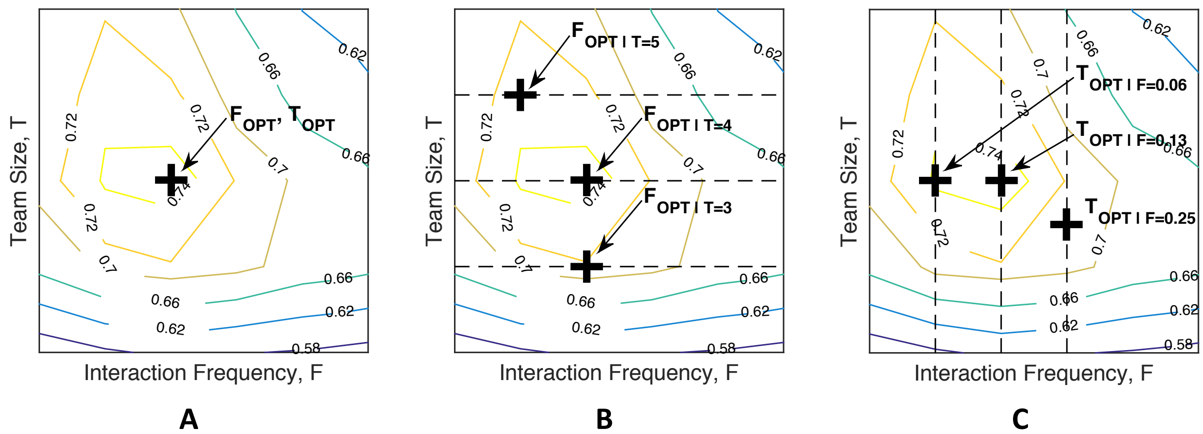


FIGURE 4.4. DETERMINING OPTIMAL TEAM CHARACTERISTICS FOR (A) CASE 1, SELECTION OF BOTH TEAM SIZE AND INTERACTION FREQUENCY, (B) CASE 2, INTERACTION FREQUENCY SELECTION WITH FIXED TEAM SIZE, AND (C) CASE 3, TEAM SIZE SELECTION WITH FIXED INTERACTION FREQUENCY.

Case 2 only requires an interaction frequency to be chosen – the team size is fixed. This allows team size to be used as a predictor variable since its values are given a priori. Case 2 uses the same information as Case 1, but multiple values of optimal interaction frequency are chosen for every combination of design problem and resource availability, one for each value of team size that was simulated (see Figure 4.4.b). This optimal interaction frequency is denoted by $F_{OPT|T}$ (optimal interaction frequency given team size) to differentiate it from F_{OPT} from Case 1. When applied across all simulations this procedure results in a data set of 160 samples (8 design problems × 4 values of R × 5 team sizes). Every observation consists of a single dependent variable ($F_{OPT|T}$) and 5 independent variables (c_A, c_G, c_L, R, and the given team size, T). This data set forms the basis of the regression analysis for Case 2.

Serving as the converse of Case 2, Case 3 only requires a team size to be chosen – the interaction frequency is fixed. This allows interaction frequency to be used as a predictor variable since its values are given a priori. Case 3 uses the same information as Cases 1 and 2, but multiple values of optimal team size are chosen for every combination of design problem and resource availability, one for each value of interaction frequency that was simulated (see Figure 4.4.c). This optimal team size is denoted by $T_{OPT|F}$ (optimal team size given interaction frequency) to differentiate it from T_{OPT} from Case 1. When applied across all simulations this procedure results in a data set of 224 samples (8 design problems × 4 values of R × 7 interaction frequencies). Every observation consists of a single dependent variable ($T_{OPT|F}$) and 5 independent variables (c_A, c_G, c_L, R, and the given interaction frequency, F). This data set forms the basis of the regression analysis for Case 3.

4.6 Regression Analysis

Regression analysis can be used to create equations that relate the properties of the design problems (computed in Section 4.4 for the design problems defined in Section 4.3) to the best team characteristics for solving those problems (elucidated in Section 4.5). The resulting regression equations are tools that can be used to organize a team to most efficiently solve a design problem.

4.6.1 Case 1: Selecting team size and interaction frequency

Case 1 addresses a scenario in which a manager or principal engineer must select both the size of the team and the frequency with which the team members interact. The task of predicting the optimal team size and the optimal frequency of interaction based on problem properties and resource limits is given mathematically by:

$$\hat{T}_{OPT} = f(c_A,\ c_G,\ c_L,\ R), \tag{4.4}$$

and

$$\hat{F}_{OPT} = f(c_A,\ c_G,\ c_L,\ R), \tag{4.5}$$

where $\hat{T}_{OPT}$ is the predicted optimal team size and $\hat{F}_{OPT}$ is the predicted optimal frequency of interaction. The functional relationship between these predicted variables and the problem properties can be determined using regression techniques. The Pearson correlation between the measured values of P_{OPT} and T_{OPT} is small and not statistically significant (Pearson $\rho = 0.167$, $p > 0.25$). Therefore a true multivariate method like reduced rank regression [136] is not required. Equations to predict optimal interaction and team size can be found separately using ordinary least-squares regression.

Main effects models were computed for predicting both team size and interaction frequency, based on the local structure property (c_L), global structure property (c_G), alignment property (c_A), and availability of resources (R). The model for optimal team size explains only 14% of the variance in T_{OPT} ($R^2_{adj} = 0.140, F = 2.26, p < 0.1$), indicating that linear main effects alone cannot effectively predict optimal team size. The model for optimal interaction frequency explains more than 70% of the variance ($R^2_{adj} = 0.729$, $F = 21.9$, $p < 0.001$), indicating that it is possible to predict optimal interaction frequency using a simple linear model.

The contribution of a term to the accuracy of the model can be assessed by defining a new model that omits that term. Comparing the accuracy of the new model to that of the complete model indicates the contribution of the omitted term. Figures 4.5 and 4.6 show the contributions from each term in the two models, computed in this fashion. The significance level of the term, computed by a t-test comparing the coefficient to 0, is indicated by asterisks. A single asterisk indicates $p < 0.1$, two asterisks indicate $p < 0.05$, and three asterisks indicate $p < 0.01$. Bars that are yellow (with a dotted edge) indicate terms that have a positive relationship with the dependent variable (i.e., increasing one will increase the other), and bars that are blue (with a solid edge) indicate a negative relationship (i.e., decreasing one will increase the other).

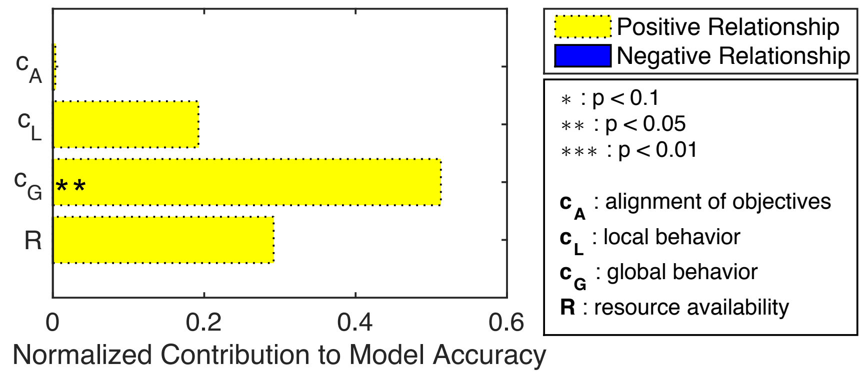


FIGURE 4.5. CONTRIBUTION TO TEAM SIZE MODEL FOR CASE 1 (SELECTION OF BOTH TEAM SIZE AND INTERACTION FREQUENCY), MAIN EFFECTS ONLY.

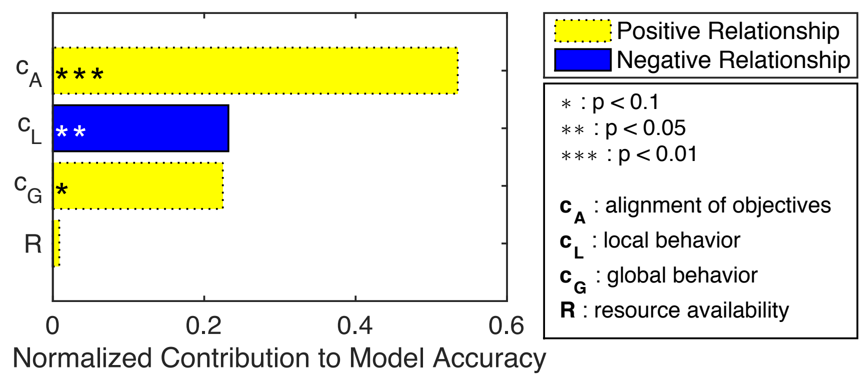


FIGURE 4.6. CONTRIBUTION TO INTERACTION FREQUENCY MODEL FOR CASE 1 (SELECTION OF BOTH TEAM SIZE AND INTERACTION FREQUENCY), MAIN EFFECTS ONLY.

The contributions in Figure 4.5 are generally less significant than those in Figure 4.6 because the team size model has lower overall accuracy (explaining only 14% of the variance in the data). Together these figures indicate that both models receive much of their accuracy from the local and global structure properties. Further, the team size model owes much of its accuracy to the resource availability variable (R), while the interaction frequency model receives a substantial contribution from the alignment variable (c_A).

The accuracy of these models can be improved by adding terms to account for interaction between variables. After adding interaction terms, the model for optimal team size explains over 50% of the variance in T_{OPT} ($R^2_{adj} = 0.513, F = 4.27, p < 0.01$), representing a significant improvement compared to the accuracy achieved with main effects alone. The elaborated model for optimal interaction frequency explains 80% of the variance ($R^2_{adj} = 0.820$, $F = 15.1$, $p < 0.001$). This only slightly improves the accuracy of the initial model, which explained 70% of the variance. Figures 4.7 and 4.8 shows the contribution to model accuracy from each term in these models after the addition of interaction terms.

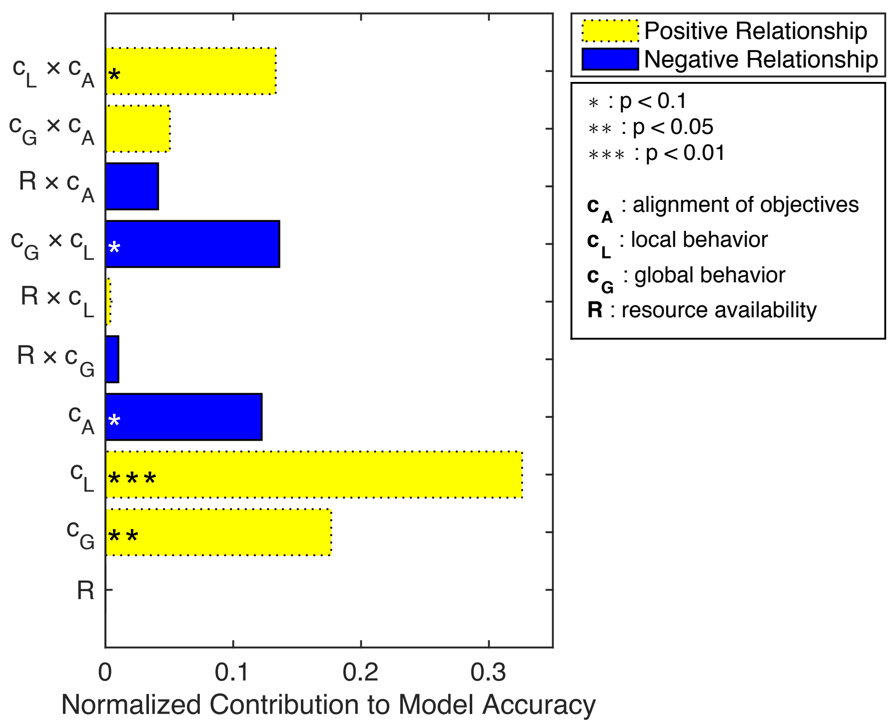


FIGURE 4.7. CONTRIBUTION TO TEAM SIZE MODEL FOR CASE 1 (SELECTION OF BOTH TEAM SIZE AND INTERACTION FREQUENCY), MAIN EFFECTS PLUS INTERACTIONS.

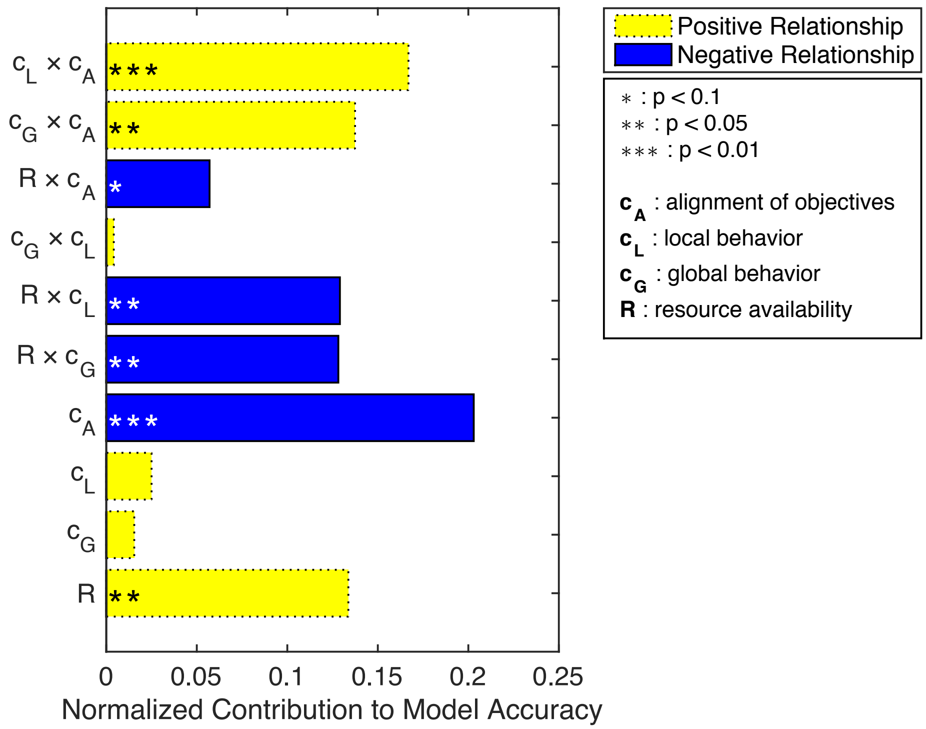


FIGURE 4.8. CONTRIBUTION TO INTERACTION FREQUENCY MODEL FOR CASE 1 (SELECTION OF BOTH TEAM SIZE AND INTERACTION FREQUENCY), MAIN EFFECTS PLUS INTERACTIONS.

Figure 4.7 indicates that the increase in accuracy of the team size model is primarily due to the interaction between local and global structure ($c_G \times c_L$) and the interaction between local structure and objective alignment ($c_L \times c_A$). Main effects of local and global structure still contribute substantially to the model's accuracy as well. Although the resource availability variable had little effect in the main effects model for interaction frequency, Figure 4.8 shows that the inclusion of interaction terms increased the contribution from that variable (both in the form of interaction terms and as a main effect).

4.6.2 Case 2: Selecting interaction frequency with team size fixed

Case 2 addresses situations in which an existing team must address a design problem. In this situation, the team size is fixed, but the frequency with which the design team interacts can be chosen by the design team manager. The task of predicting the optimal interaction frequency in this situation is given mathematically as:

$$\hat{F}_{OPT|T} = f(c_A,\ \ c_G,\ \ c_L,\ \ R,\ \ T), \tag{4.6}$$

where T is the given size of the team, which is now known a priori. An equation to predict optimal interaction frequency can be found using least-squares regression. As in Case 1, only main effects are initially considered in the model. This simple model explains over 70% of the observed variance ($R^2_{adj} = 0.726, F = 85.4, p < 0.001$), indicating once again that the best interaction frequency for solving a problem can be predicted well using only main effects (specifically the main effects for local structure, global structure, and objective alignment). The contributions from each term in the model are shown in Figure 4.9. These results are very similar to the initial model for interaction frequency in Case 1 (see Figure 4.5) with only a small contribution from the known team size.

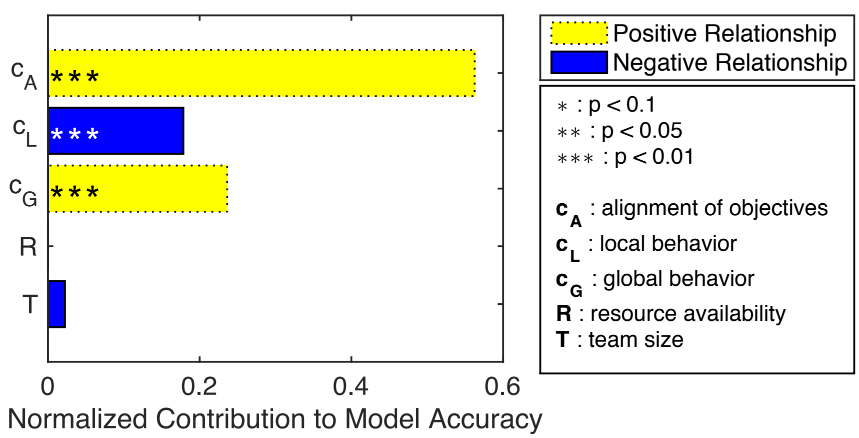


FIGURE 4.9. CONTRIBUTION TO FINAL MODEL FOR CASE 2 (INTERACTION FREQUENCY SELECTION WITH FIXED TEAM SIZE), MAIN EFFECTS ONLY.

Next, the initial model is elaborated by adding interaction terms to account for the interaction between variables. This elaborated model explains over 80% of the observed variance in P_{OPT} ($R^2_{adj} = 0.825, F = 51.1, p < 0.001$). Although the accuracy increases significantly with the inclusion of interaction terms, the model complexity also increases. The contribution from each term in the model is provided in Figure 4.10.

Including interaction terms increases the accuracy of this model by approximately 10%. While many of the interaction terms contribute to this boost in accuracy, the primary increase results from the interaction of local structure with objective alignment ($c_L \times c_A$), and global structure with objective alignment ($c_G \times c_A$). These interaction terms also contribute significantly to the interaction effects model for interaction frequency in Case 1 (see Figure 4.7).

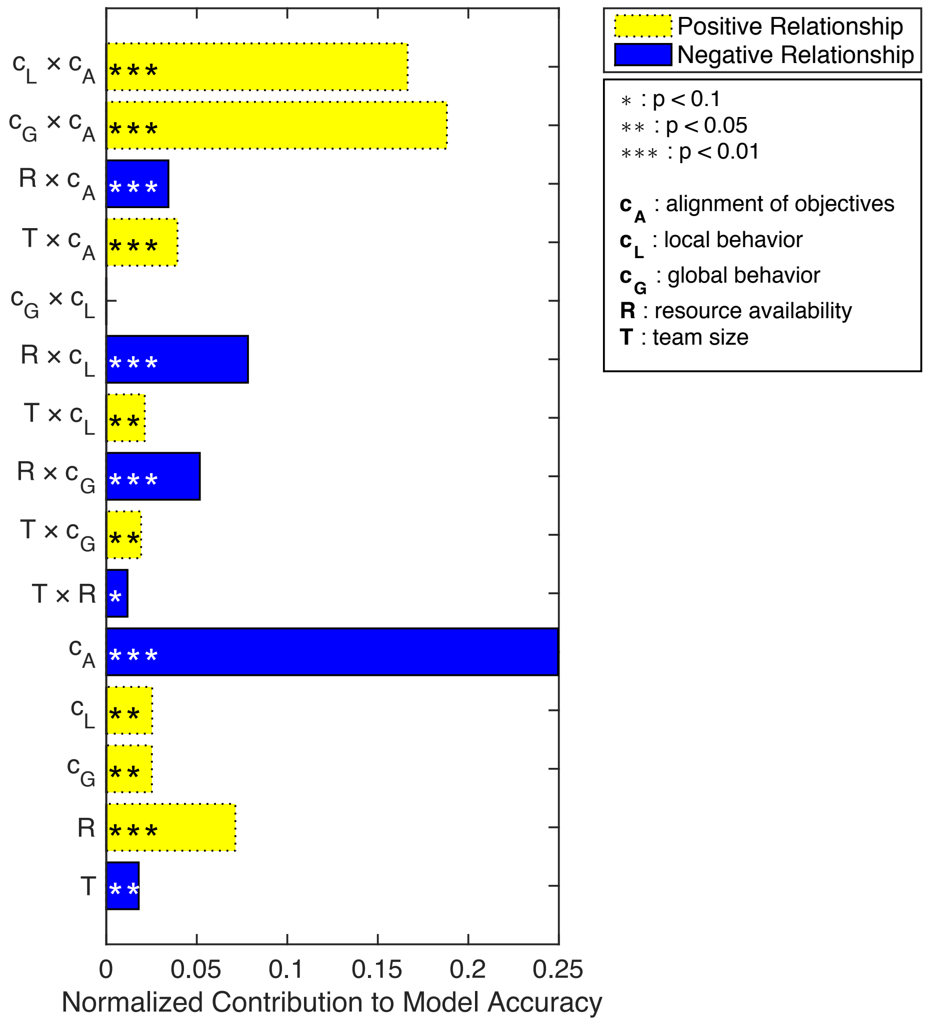


FIGURE 4.10. CONTRIBUTION TO FINAL MODEL FOR CASE 2 (INTERACTION FREQUENCY SELECTION WITH FIXED TEAM SIZE), MAIN EFFECTS PLUS INTERACTIONS.

4.6.3 Case 3: Selecting team size with interaction frequency fixed

Case 3 addresses situations in which the members of a team must adhere to a set meeting schedule. In this situation, the interaction frequency is fixed, but the size of the team can be chosen by the design team manager. The task of predicting the optimal team size in this situation is given mathematically as:

$$\hat{T}_{OPT|F} = f(c_A,\ c_G,\ c_L,\ R,\ F), \qquad 4.7$$

where F is the given interaction frequency, which is now known a priori. An equation to predict optimal team can be found using least-squares regression. As in the first two cases, we first include only main effects in the model. Main effects are capable of explaining approximately 25% of the observed variance ($R^2_{adj} = 0.259, F = 16.6, p < 0.001$). This is higher accuracy than the initial main effects model for team size in Case 1, indicating that knowledge of interaction frequency imparts some additional predictive power. The contributions from each term in this main effects model are shown in Figure 4.11. Much of the predictive power of this model is derived from knowledge of the global structure of the space and available resources, much like the team size main effects model for Case 1 (see Figure 4.5).

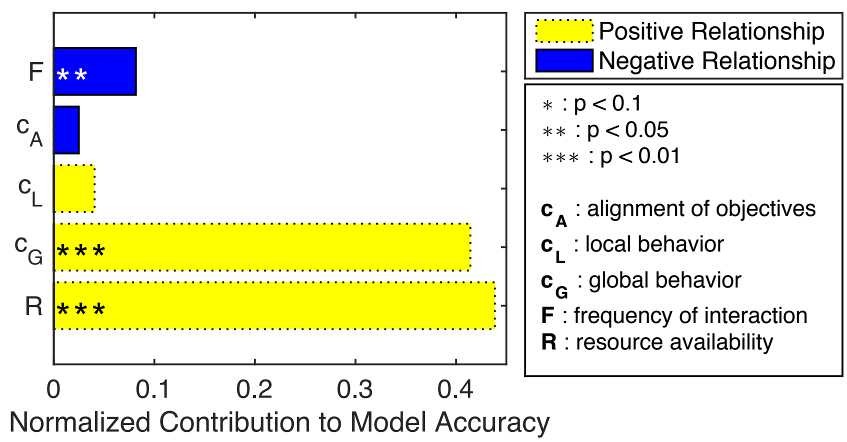


FIGURE 4.11. CONTRIBUTION TO FINAL MODEL FOR CASE 3 (TEAM SIZE SELECTION WITH FIXED INTERACTION FREQUENCY), MAIN EFFECTS ONLY.

Next, the main effects model is extended by adding interaction terms to account for the interaction between variables. This elaborated model explains over 80% of the observed variance in P_{OPT} ($R^2_{adj} = 0.545, F = 18.8, p < 0.001$). Although the accuracy increases significantly with the inclusion of interaction terms, the model complexity also increases. The contribution from each term in the model is provided in Figure 4.12.

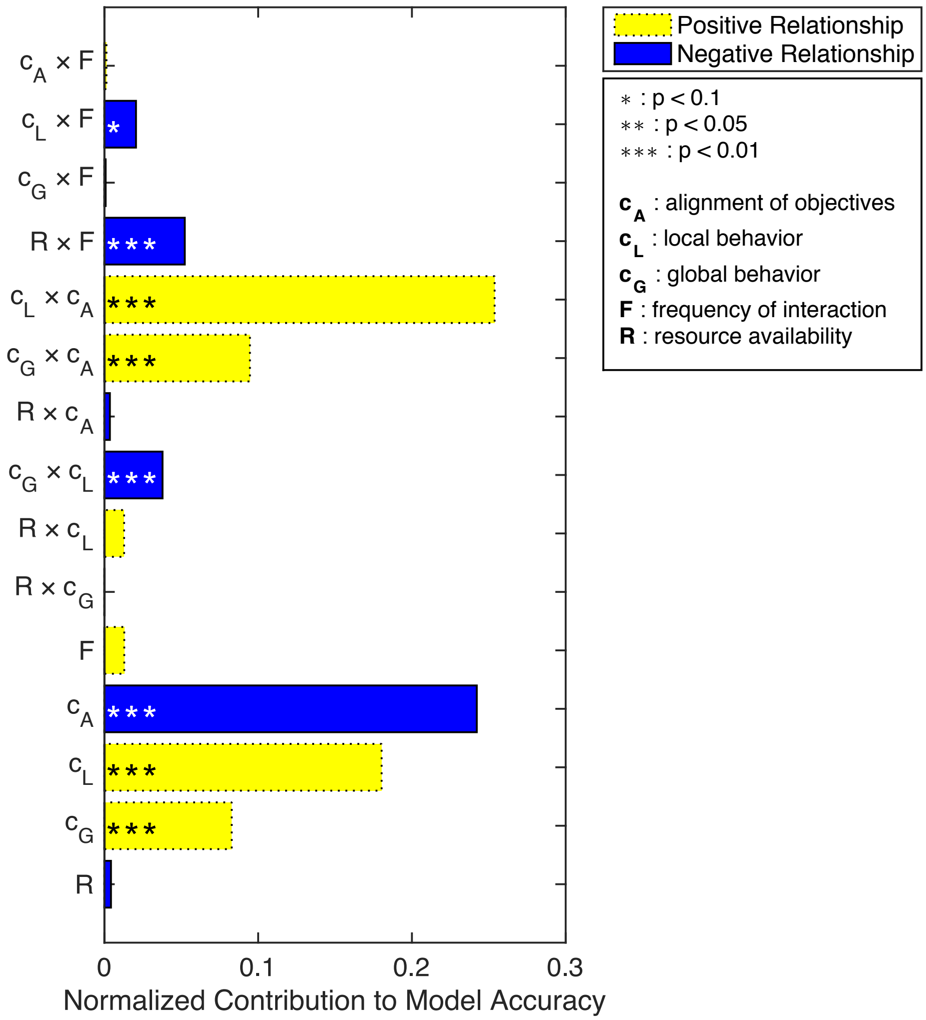


FIGURE 4.12. CONTRIBUTION TO FINAL MODEL FOR CASE 3 (TEAM SIZE SELECTION WITH FIXED INTERACTION FREQUENCY), MAIN EFFECTS PLUS INTERACTIONS.

Including interaction terms increases the accuracy of this model by nearly 30%. Many of the interaction terms contribute to this boost in accuracy with the largest contributions resulting from the interaction of local structure with objective alignment ($c_L \times c_A$), and global structure with objective alignment ($c_G \times c_A$). These interaction terms also contribute significantly to the interaction effects model for interaction frequency in Cases 1 and 2 (see Figures 4.8 and 4.10) as well as the interaction effects model for team size from Case 1 (see Figure 4.7). A significant contribution to model accuracy is also derived from main effects for objective alignment (c_A), local structure (c_L), and global structure (c_G).

4.7 Discussion

Table 4.1 provides a summary of the regression analyses conducted in this chapter. Case 1 addresses the independent prediction of optimal team size and optimal interaction frequency; Case 2 addresses the prediction of optimal interaction frequency for a given team size; and Case 3 addresses the prediction of optimal team size for a given interaction frequency. For each prediction task, the model was initialized with only main effects, and interaction terms were later added to account for the potential interplay between variables.

Models for predicting optimal team size were computed in Case 1 and Case 3. The cases are differentiated by the fact that interaction frequency is used as an independent variable in Case 3, but not in Case 1. In Case 1, the inclusion of interaction effects more than triples the accuracy of the model, and nearly doubles accuracy in Case 3 (see Figures 4.5 and 4.11 for main effects models, and Figures 4.7 and 4.12 for models after the inclusion of interaction terms). The inclusion of interaction effects also greatly increases the complexity of the models, adding 6 more terms in Case 1 and 10 terms in Case 3. The stark increase in model accuracy justifies the increased model complexity, so the elaborated versions of the models (main effects plus interaction effects) are preferred. These models are noted in Table 4.1.

Models for predicting optimal interaction frequency were computed in Case 1 and Case 2, with the sole difference being that team size is used as an independent variables in Case 3. The models achieve high statistical significance and good accuracy both before and after the inclusion fo interaction terms. Adding interaction terms adds 6 additional terms to the regression equation in Case 1 and 10 additional terms in Case 1, more than doubling the total number of terms in the model (see Figures 4.6 and 4.9 for the main effects models, and Figures 4.8 and 4.10 for the models after the inclusion of interaction terms). This large increase in complexity only boosts the accuracy of the model by about 10% for either case. In addition, the larger number of terms makes the model more challenging to interpret and greatly increases the likelihood of overfitting. Therefore, the initial models with main effects only are preferred when predicting interaction frequency. These preferred models are also noted in Table 4.1.

TABLE 4.1. SUMMARY OF REGRESSION MODELS.

Case	Independent variable	Terms	R^2_{adj}	p-value	Preferred
1	T_{OPT}	main effects only	0.140	< 0.1	No
		main effects + interactions	0.513	< 0.01	Yes (see Table 4.2)
	F_{OPT}	main effects only	0.729	< 0.001	Yes (see Table 4.3)
		main effects + interactions	0.820	< 0.001	No
2	$F_{OPT\|T}$	main effects only	0.726	< 0.001	Yes (see Table 4.4)
		main effects + interactions	0.825	< 0.001	No
3	$T_{OPT\|F}$	main effects only	0.259	< 0.001	No
		main effects + interactions	0.545	< 0.001	Yes (see Table 4.5)

Additional information pertaining to the preferred models identified in Table 4.1. There models are the preferred team size model for Case 1 (given in Table 4.2), the preferred interaction frequency model for Case 1 (given in Table 4.3), the preferred interaction frequency model for Case 2 (given in Table 4.4), and the preferred team size model for Case 3 (given in Table 4.5). Each model is broken down according to the terms that compose that model. For every term a description is given, following by the estimated coefficient of that term, the standard error associated with the estimate (S.E.), the Student's test statistic for a comparison of the estimated coefficient to 0 (t), and the significance level according to that test statistic (p).

TABLE 4.2. COEFFICIENT VALUES AND STATISTICS OF PREFERRED TEAM SIZE MODEL FOR CASE 1 (SELECTION OF BOTH TEAM SIZE AND INTERACTION FREQUENCY).

Term	Term description	Estimate	S.E.	t	p
Intercept	Intercept	-2.10×10^{2}	7.10×10^{1}	-2.95×10^{0}	< 0.01
R	Availability of resources	-6.08×10^{-4}	1.39×10^{-2}	-4.37×10^{-2}	> 0.1
c_G	Global structure	3.69×10^{2}	1.56×10^{2}	2.36×10^{0}	< 0.05
c_L	Local structure	1.66×10^{2}	5.16×10^{1}	3.21×10^{0}	< 0.01
c_A	Alignment of objectives	-1.25×10^{2}	6.35×10^{1}	-1.97×10^{0}	< 0.1
$R \times c_G$	Interaction: availability of resources, global structure	-7.44×10^{-3}	1.31×10^{-2}	-5.67×10^{-1}	> 0.1
$R \times c_L$	Interaction: availability of resources, local structure	3.04×10^{-3}	8.69×10^{-3}	3.50×10^{-1}	> 0.1
$c_G \times c_L$	Interaction: global structure, local structure	-2.63×10^{2}	1.27×10^{2}	-2.07×10^{0}	< 0.1
$R \times c_A$	Interaction: availability of resources, alignment of objectives	-2.45×10^{-3}	2.15×10^{-3}	-1.14×10^{0}	> 0.1
$c_G \times c_A$	Interaction: global structure, alignment of objectives	1.06×10^{2}	8.42×10^{1}	1.26×10^{0}	> 0.1
$c_L \times c_A$	Interaction: local structure, alignment of objectives	7.77×10^{1}	3.79×10^{1}	2.05×10^{0}	< 0.1

TABLE 4.3. COEFFICIENT VALUES AND STATISTICS OF PREFERRED INTERACTION FREQUENCY MODEL FOR CASE 1 (SELECTION OF BOTH TEAM SIZE AND INTERACTION FREQUENCY).

Term	**Term description**	**Estimate**	**S.E.**	t	p
Intercept	Intercept	-2.10×10^{2}	7.10×10^{1}	-2.95×10^{0}	> 0.1
R	Availability of resources	-6.08×10^{-4}	1.39×10^{-2}	-4.37×10^{-2}	> 0.1
c_G	Global structure	3.69×10^{2}	1.56×10^{2}	2.36×10^{0}	< 0.1
c_L	Local structure	1.66×10^{2}	5.16×10^{1}	3.21×10^{0}	< 0.05
c_A	Alignment of objectives	-1.25×10^{2}	6.35×10^{1}	-1.97×10^{0}	< 0.01

TABLE 4.4. COEFFICIENT VALUES AND STATISTICS OF PREFERRED INTERACTION FREQUENCY MODEL FOR CASE 2 (INTERACTION FREQUENCY SELECTION WITH FIXED TEAM SIZE).

Term	**Term description**	**Estimate**	**S.E.**	t	p
Intercept	Intercept	2.00×10^{0}	1.03×10^{0}	1.95×10^{0}	< 0.1
T	Team size	-1.95×10^{-2}	1.33×10^{-2}	-1.47×10^{0}	> 0.1
R	Availability of resources	-4.06×10^{-6}	3.36×10^{-5}	-1.21×10^{-1}	> 0.1
c_G	Global structure	4.58×10^{0}	9.63×10^{-1}	4.76×10^{0}	< 0.01
c_L	Local structure	-2.64×10^{0}	6.38×10^{-1}	-4.14×10^{0}	< 0.01
c_A	Alignment of objectives	1.16×10^{0}	1.58×10^{-1}	7.35×10^{0}	< 0.01

TABLE 4.5. COEFFICIENT VALUES AND STATISTICS OF PREFERRED TEAM SIZE MODEL FOR CASE 3 (TEAM SIZE SELECTION WITH FIXED INTERACTION FREQUENCY).

Term	Term description	Estimate	S.E.	t	p
Intercept	Intercept	-1.83×10^{2}	3.06×10^{1}	-5.97×10^{0}	< 0.01
R	Availability of resources	-5.36×10^{-3}	5.98×10^{-3}	-8.96×10^{-1}	> 0.1
c_G	Global structure	2.64×10^{2}	6.70×10^{1}	3.95×10^{0}	< 0.01
c_L	Local structure	1.29×10^{2}	2.22×10^{1}	5.83×10^{0}	< 0.01
c_A	Alignment of objectives	-1.84×10^{2}	2.72×10^{1}	-6.75×10^{0}	< 0.01
F	Frequency of interaction	1.56×10^{1}	1.00×10^{1}	1.56×10^{0}	> 0.1
$R \times c_G$	Interaction: availability of resources, global structure	-2.02×10^{-4}	5.63×10^{-3}	-3.60×10^{-2}	> 0.1
$R \times c_L$	Interaction: availability of resources, local structure	5.80×10^{-3}	3.73×10^{-3}	1.55×10^{0}	> 0.1
$c_G \times c_L$	Interaction: global structure, local structure	-1.45×10^{2}	5.43×10^{1}	-2.67×10^{0}	< 0.01
$R \times c_A$	Interaction: availability of resources, alignment of objectives	-7.55×10^{-4}	9.22×10^{-4}	-8.19×10^{-1}	> 0.1
$c_G \times c_A$	Interaction: global structure, alignment of objectives	1.53×10^{2}	3.61×10^{1}	4.22×10^{0}	< 0.01
$c_L \times c_A$	Interaction: local structure, alignment of objectives	1.12×10^{2}	1.62×10^{1}	6.91×10^{0}	< 0.01
$R \times F$	Interaction: availability of resources, interaction frequency	-1.03×10^{-3}	3.29×10^{-4}	-3.14×10^{0}	< 0.01
$c_G \times F$	Interaction: availability of resources, global structure	-3.64×10^{0}	9.43×10^{0}	-3.86×10^{-1}	> 0.1
$c_L \times F$	Interaction: availability of resources, local structure	-1.23×10^{1}	6.25×10^{0}	-1.97×10^{0}	< 0.1
$c_A \times F$	Interaction: availability of resources, alignment of objectives	7.77×10^{-1}	1.54×10^{0}	5.03×10^{-1}	> 0.1

4.7.1 Predicting Optimal Team Size

The preferred models for predicting optimal team size (from Case 1 and Case 3) contain interaction effects in addition to main effects. Both models explain more than half of the observed variance, indicating moderate accuracy, and also achieve high statistical significance. Comparing Figures 4.7 and 4.12 indicates that the preferred models for the two cases are also functionally similar. These figures specifically show that the terms corresponding to the alignment of objectives (c_A), local structure (c_L), and the interaction between them structure ($c_L \times c_A$), contribute substantially to both team size models.

The objective alignment property is negatively related to team size, indicating that optimal team size decreases as objective functions become more aligned. For problems that contain unaligned objective functions, it is likely that larger team sizes allow for search patterns that are more divergent in nature. This, in turn, enables the team to more readily search for a portion of the design space in which objectives are aligned or in which all objectives reach acceptable values. The local structure property is positively related to team size, which indicates that optimal team size increases as the design space becomes more locally rough. Rougher design spaces are more difficult to search, so increasing the size of a team allows for greater search breadth. The interaction term between alignment and local structure is positively correlated with team size, meaning that small changes in these design space properties have the largest impact when the design space is aligned and smooth (that is, when c_A is high and c_L is low).

The availability of resources (R) contributes substantially to the accuracy of all models for predicting optimal team size, either as a main effect or through interacting with other variables. As a main effect in Case 1 and Case 3, the term corresponding to availability of resources is positively related to optimal team size. In other words, when more resources are available it is beneficial to increase team size, spreading resources amongst a greater number of individuals to increase the extent to which work can be done concurrently. This corresponds to the conventional wisdom for smaller software development teams [22,115,116]. This indicates that it may be possible to successfully apply some of the best practices from software development to the design of mechanical systems, especially if design tasks are similar to those used in this work.

This work shows that optimal team size is significantly related to several design problem properties, but the best models for team size only explain slightly more than half of the variance in the data. One possible cause for this large residual is that the relationship between the design problem properties and optimal team size cannot be adequately captured with linear models, necessitating higher order methods. Another possibility is that one or more design problems properties that were not included in this work actually play a significant role in driving optimal team size.

4.7.2 Predicting Optimal Interaction Frequency

An inspection of Figures 4.6 and 4.9 indicates that the preferred models for interaction frequency are very similar. This is expected since the models only differ in the addition of team size as a predictor, and team size was shown to only have a small correlation with interaction frequency for the team sizes investigated in this work. In both of these preferred models the design space properties (c_A, c_G, and c_L) are the most impactful terms, while the resource availability variable (R) has no significant predictive value. Further, when knowledge of the size of the team (T) is available it does not contribute significantly to the predictive ability of the relationship.

The objective alignment property is positively related to interaction frequency. In other words, less frequent interaction is preferred for design problems in which objective functions are not aligned. If objective functions are not aligned, then infrequent interaction allows for divergent search to take place as members of the team look for regions of the design space in which all objective functions reach suitable values.

The local structure property is negatively related to optimal interaction frequency, which indicates that more frequent interaction is only beneficial if the design space is locally smooth. If the design space is locally rough, infrequent interaction enables individuals to perform diligent local search before considering trade-offs between solutions. Interacting too often could cause premature convergence in a portion of the design space that does not contain satisfactory solutions.

Finally, the global structure property is positively correlated with optimal interaction frequency, indicating that multimodal design spaces require less frequent interaction. The reasoning for this relationship is much the same as that for the objective alignment and local structure properties. Infrequent interaction allows individuals to act independently as they individually find local minima. Once individuals in a team have found a diverse set of local minima, interacting allows them to select a shrinking set of alternatives to pursue.

Considering the extreme values of the design space properties provides two illustrative examples of their relationship to optimal interaction frequency. On one hand, infrequent interaction is preferred for design problems in which objective functions are unaligned, exhibit rough local structure, and are highly multimodal. This enables individuals to spend time independently refining solutions (essentially finding a set of local minima within the team) before the team interacts to consider trade-offs between the solutions in the set. In this scenario, frequent interaction could lead to premature convergence on a poor local minimum. On the other hand, frequent interaction is desirable for a design space in which objectives are aligned, there are few local minima, and the objective functions are locally smooth. This would enable a team of individuals to rapidly converge on a solution without spending undue time on divergent search.

Other work has studied the relationship between project complexity and task performance [21]. In that work, project complexity was measured through ambiguity (comparable to the inverse of this work's alignment variable, c_A) and multiplicity (similar to the inverse of this work's global structure variable, c_G). That work identified nontrivial interaction between multiplicity and ambiguity, a result that is echoed in the interaction between the global structure and objective alignment variables ($c_A \times c_G$) in this work (see Figures 4.8 and 4.10).

4.7.3 Generalization and Limitations

Care was taken to ensure that each of the design problem properties could be computed before solving begins using a random walk procedure. This procedure makes two important assumptions about how the design problem is formulated. These two assumptions must be true in order to compute the design problem properties defined in this chapter, and these properties are in turn prerequisite for computing optimal team characteristics. More broadly, these two assumptions must be true to simulate a problem using the CISAT framework.

The first assumption is that the quality of a solution can be quantified with one or more well-defined objective functions. It may not always be feasible to define numerical objective functions for a design problem, especially if the objectives of the design are subjective in nature. However, it is possible to employ user surveys to robustly quantify subjective criteria like elegance or sportiness [137]. Such ratings-based data could be used to compute properties of the design problem, and from those properties the optimal team characteristics could be estimated.

The second assumption is that existing solutions can be modified efficiently, making it possible to produce a random walk through the solution space with little effort. This assumption was addressed here by ensuring that well-defined rules for modifying solutions were associated with each design problem, allowing the random walk procedure to be automated. It is straightforward to produce a random walk for any design problem for which such rules exist, such as problems defined by design grammars. Design grammars have been developed for a broad array of design problems, including the synthesis of gear trains [138–140], neural networks [141], function structures [142], photovoltaic arrays [143], and various structural systems [90,93,144]. Random walks through the design space are sometimes used during the grammar development process as a means of better understanding and refining the rules of the design grammar [145]. Thus, it may be possible to compute the design problem properties introduced here during grammar development, enabling the simultaneous refinement of grammar rules and suggested team solving approaches.

It may be possible to estimate problem properties for some design problems. For instance, it may be possible to infer an approximate value for the alignment of objectives (c_A) based on known relationships between objectives. In the example used previously, it may be easy to recognize that there is high positive alignment between objectives to minimize mass and minimize material costs. Similarly, it is usually the case that the cost of a part or product is inversely related to the strength of the product, indicating a negative value for c_A. Such qualitative insights could be used to inform the selection of good team characteristics in the absence of a more comprehensive evaluation of the design problem. It may be possible to develop quantitative guidelines for estimating other properties as well.

The CISAT modeling framework is necessary to evaluate the large number of different conditions considered in this chapter – conducting a similar project using behavioral studies alone would not be feasible. However, the computational expediency of CISAT brings with it several limitations. For instance, CISAT only models a limited (albeit important) set of phenomena. The influence on the performance of engineering design teams of phenomena that are not modeled could be significant. Further, this work focuses on the impact of team size and interaction frequency, while holding other characteristics constant (like self-bias and satisficing behavior, two characteristics that *can* be modulated in the CISAT framework). It is possible that these characteristics might interact with team size and interaction frequency to influence team performance.

4.8 Summary

This chapter used the CISAT modeling framework to define the relationship between the properties of design problems and the team characteristics that lead to the best solutions to those problems. The CISAT framework was a necessary replacement for traditional cognitive studies because of the sheer breadth of cases that had to be considered. The simulation results presented in this chapter represent over 100,000 hours of equivalent human design studies..

First, three design problem properties were defined and used to quantify design problems with. These properties were the local structure of the design space, the global structure, and the extent to which objectives are aligned. These design properties can be computed before solving begins using a random walk procedure. The CISAT modeling framework was then used to simulate the performance of engineering design teams with a broad variety of team characteristics on several different engineering design problems. Post-processing of these results provided a set of optimal team characteristics for each of the design problems. Finally, regression analysis was used to define equations relating the optimal team characteristics to properties describing the design problems. These equations make it possible to predict optimal design team characteristics (team size and interaction frequency) based on design problem properties. Because properties of the design problem can be computed before solving begins, these predictive equations are powerful tools that facilitate the optimal design of design teams.

The selection of the optimal number of individuals in a team is a complicated relationship, depending greatly on the design space properties as well as the interactions between them. In addition, the availability of resources plays a large role in the selection of an optimal team size The selection of an optimal interaction frequency can be predicted with high accuracy based on the main effects of design space properties without the need to consider interaction effects. If a design problem has unaligned objectives, rough local structure, and is highly multimodal, then teams should interact infrequently and spend time working independently to avoid premature convergence on unacceptable solutions. In contrast, if a design problem has aligned objectives, smooth local structure, and fewer local minima, then frequent interaction within the team can yield a quick search that converges on an acceptable solution.

This chapter focused on the selection of optimal team size and optimal interaction frequency, but the approach used here of exploring team characteristics using computational simulations of human teams could be applied to a variety of additional team characteristics as long as they are manipulable through the CISAT framework. The approach used here of informing team characteristics based on design problem properties could also be used to develop guidelines for the selection of optimally-directed parameters for computational design algorithms as well.

Future work should continue to refine the relationships defined here by simulating design team performance on additional problems and exploring new design space properties (especially those that might drive optimal team size). Future work should also explore the development of similar relationships for teams employing hierarchical structures or consisting of individuals with highly-specialized skills or knowledge. The range of design problem types (conceptual design, topology design, and detailed design) to which these results apply needs to be examined. Such extensions of this work have the potential to develop a deeper and richer understanding of the search process as it applies to both humans and algorithms.

This chapter used the CISAT modeling framework to conduct a host of simulations, eventually uncovering relationships that theoretically allow the best characteristics for solving a design problem to be predicted based on measured values of the problem's properties. The next chapter presents the results of a cognitive study of human design teams that provides a limited validation of the computationally derived theories presented in this chapter.

Chapter 5: Optimizing Design Teams: Limited Theory Validation via Cognitive Study

Knowing is not enough; we must apply. Willing is not enough; we must do.

Johann Wolfgang von Goethe

5.1 Overview

Teams are not necessarily a panacea for all design problems; as demonstrated in Chapter 4, team effectiveness is highly dependent on how the team is structured, how individuals in the team communicate, and the properties of the problem being solved. Yet it is generally assumed in practice and research that interacting teams are more effective than individuals across a variety of tasks. This assumption is not unfounded – teams have been shown to have superior performance in computer-facilitated idea generation [146,147], and concept evaluation and selection [14]. Some practitioners even propose that with the right instructions teams can always be more effective than individuals [13].

In Chapter 4, an array of computational simulations of engineering design teams were conducted to develop a set of relationships for predicting the optimal values for team characteristics. The current chapter seeks to provide a limited validation for those predictive equations through a cognitive study. We specifically target the relationships for optimal interaction frequency since they achieved the highest coefficients of correlation and thus are likely to be most impactful.

This chapter first introduces the design task that was solved by participants during the cognitive study: a challenging configuration problem of moderate size that requires single-domain knowledge. Next, we measure the properties of that design problem and use the property values to estimate the problem-specific optimal interaction frequency for teams to use. For the configuration problem used here, we predicted (rather surprisingly) that zero interaction between team members should lead to the best outcome. In other words, team members should work as individuals and the best solution should be selected upon completion. Small teams of engineering students solved the design problem, either using the predicted optimal interaction frequency (zero interaction) or one of two non-zero frequencies. The results from this study support the surprising prediction. Further investigation of participants' perceptions and activity during design reveals both benefits and detriments that emerged as a result of using the optimal interaction frequency.

5.2 Background

In the study of team interaction, the most overt comparison to draw is between nominal teams (in which individuals work independently) and interacting teams (in which communication between members is allowed). Results have been mixed, and it appears that the existence of teammates may either augment or interfere with an individual's cognitive processes [148,149]. For instance, nominal teams tend to produce concepts in greater quantity and with higher quality than interacting teams when completing concept generation tasks [15,16]. This result is attributed to production blocking in interacting teams, or the fact that team members must take turn speaking [150,151]. Poor performance in interacting teams can also result from social loafing, which is the tendency of an individual to exert less effort when working as part of a team [20]. However, there exist cases in which interacting teams may outperform nominal teams. For instance, teams collaborating through an electronic medium can outperform both nominal and face-to-face interacting teams on brainstorming tasks [146,147]. It has also been demonstrated that teams are usually better than individuals when evaluating potential solutions [14]. Katezenbach and Smith even go so far as to posit that interacting teams will always outperform nominal teams when the teams are appropriately supported and operated [13].

Comparative research on interacting versus nominal (aka non-interacting) teams treats communication as a binary variable – interaction between team members is either prohibited or allowed. However, this ignores the reality that communication within teams can vary continuously across a variety of dimensions. Interaction frequency is commonly used as a measure of the total amount of communication within a team [119–122]. It has been shown that the relationship between frequency of interaction and task performance is approximately quadratic in cross-functional teams [123]. For most problem types, high and low interaction frequencies result in lower performance, with a well-defined optimum somewhere in between [123]. A computational model was developed to investigate this phenomenon further, and it indicated that higher interaction frequency tended to decrease the quality of communication [124]. Further computational work demonstrated a relationship between optimal interaction frequency and project complexity [21]. In the software development literature, Brooks' Law describes the relationship between the optimal team size and performance as a function of the level of communication overhead required for the task [115,116]. The communication structure (or lack thereof) that is used in software teams can also significantly impact effectiveness. It has been shown that the optimal choice of communicative structure is heavily task dependent [152].

Other work has looked beyond the effects of interaction on task performance to assess the degree to which communication can impact individual well-being. Face-to-face interaction in large organizations is positively correlated with job satisfaction and the positive perception of interaction quality [153]. However, increasing the number of scheduled meetings can increase fatigue and also increase perceived workload [154]. The key difference between these findings is the naturally scheduled versus interrupting nature of face-to-face interactions and meetings, respectively. Whereas face-to-face interaction can occur naturally in between tasks or as a part of work, meetings are typically perceived as an interruption [154]. Interruptions tend to increases the rate of work, but they also result in a more negative mood [155,156].

5.3 Design Task

The design problem used here tasked participants with design of a system of connected products to maintain the temperature within a house. The house consisted of 13 rooms connected with doorways with a floorplan as shown in Figure 5.1.

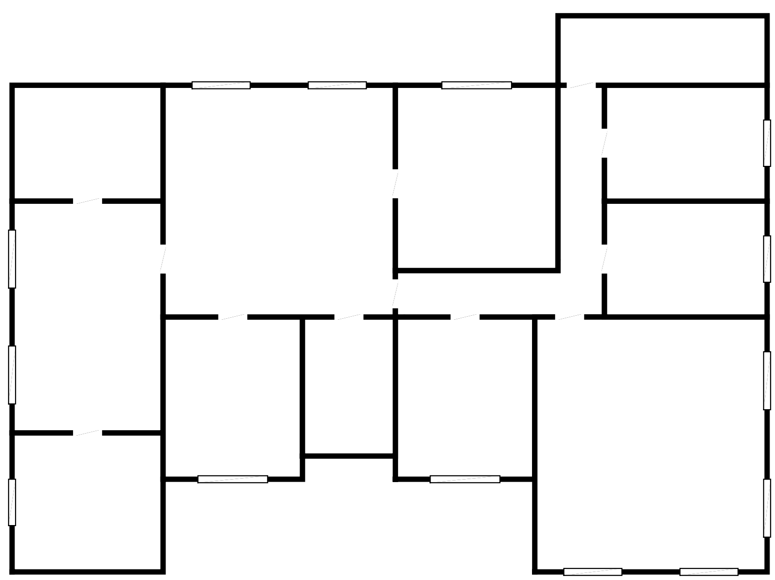


FIGURE 5.1. HOUSE LAYOUT FOR COOLING SYSTEM DESIGN TASK.

Participants were allowed to use three product types to create their solution: sensors, processors, and coolers. Sensors measured the temperature of rooms in which they were placed.

Coolers took external air and delivered it to the internal environment at a lower-than-ambient temperature. Processors provided a means for connecting sensors and coolers, taking temperature information from sensors, and deciding whether or not to turn on coolers. Processors were only capable of receiving information from or acting on products to which they were explicitly connected. Using processors, participants were able to create numerous independent subsystems as part of the same solution. In searching for an adequate solution, participants were allowed to add, delete, and move products. They were also allowed to tune the power and flowrate of coolers.

To evaluate a solution, the distribution of temperatures within the home was simulated for an average day with external temperature varying between 20°C and 30°C. The mean temperature within each room of the home was solved using principles of heat and mass transfer. Two metrics (peak temperature and total cost) were computed based on the log of temperatures and product activation for the simulation. Peak temperature was defined as the highest temperature obtained in any room in the house during the simulated time period. Total cost was computed as the sum of the cost of the products making up the system and the projected 10-year operating cost.

This design task admitted a large variety of potential solutions. This ensured that individuals within a team could pursue different designs as well as guaranteeing some degree of difference between the eventual problem solutions that teams found. It was also a challenging design task that proved difficult for participants, ensuring their continued effort over the course of the study. It should also be noted that this task only required participants to have knowledge of a single domain.

5.4 Characterization and Prediction

One hundred random walks were taken for the cooling system design problem and values for objective alignment (c_A), global behavior (c_G), and local behavior (c_L) were computed for each random walk. The average values for the properties were $c_A = -0.892 \pm 0.011$, $c_G = 0.423 \pm 0.012$, and $c_L = 1.093 \pm 0.009$. The error term in these measurements represents the standard error of the mean. Further, a team size of 3 ($T = 3$) was selected and it was determined that 50 design actions should be allowed per individual, resulting in a total of 150 actions per team ($R = 150$). These values were chosen based on the results of pilot studies with the objective of enabling all participants to complete the study within one hour.

Substituting the values for c_A, c_G, c_L, R, and T into the preferred model for Case 2 (see Chapter 4) yields a value of $\hat{F}_{OPT} = -0.036 \pm 0.057$. This numerical value represents the fraction of the team's effort that should be spent on interaction. Although the value is slightly negative, it is not significantly different from 0. One interpretation of this prediction is that non-interacting teams should provide solutions with the highest quality for this design problem. However, given the mean error of the prediction, it is possible that some small degree of interaction leads to optimal performance for this problem. Therefore, an alternative interpretation of the prediction is that any appreciably large interaction frequency should result in decreased solution quality. In either case, cognitive study results should show a strong preference towards interaction frequencies that are near or at zero. This prediction was used to inform the design of the cognitive study that is introduced in the next section.

5.5 Study Overview

The primary purpose of this cognitive study was to examine the effect of interaction frequency on team performance for the specific task and, in doing so, attempt to offer a limited validation of the optimal interaction frequency predicted in the previous section. Teams were partitioned into three conditions, and members of teams in each condition were prompted to interact at different frequencies (with instructions not to interact unless prompted). Teams in Conditions 1 were nominal teams in which individual participants worked separately (an interaction frequency of 0.0), teams in Condition 2 interacted after every 10 design actions (an interaction frequency of 0.1), and teams in Condition 3 interacted after every 5 design actions (an interaction frequency of 0.2). Based on the prediction from the previous section, teams in Condition 1 were expected to provide solutions with the highest quality, namely the team as a whole should provide the best solution when the members do not communicate with each other.

Design was facilitated with a computer interface that allowed participants to construct, evaluate, and share solutions within their team. This interface also prompted teams to interact at the correct frequency for their condition, and recorded all design actions taken by participants, enabling later reconstruction of a full account of design activity.

5.5.1 Participants

This study was conducted with senior undergraduates and graduate students in mechanical engineering. Since courses in heat transfer and thermodynamics are part of a standard mechanical engineering curriculum, it can be assumed that participants were familiar with these topics, and thus possessed a basic understanding of the physics involved in this design problem.

The median age of the participants was 22 with a range of 21-31. There were 40 male students and 14 female students, and 37 senior undergraduate students and 17 graduate students.

Of the 54 participants, 12 were allocated to Condition 1, 21 were allocated to Condition 2, and 21 were allocated to Condition 3. Conditions 2 and 3 required participants to work collaboratively in teams, but Condition 1 required participants to work independently. Since the activities of individual participants in Condition 1 were statistically independent of one another, later analysis considers all possible team combinations that could be assembled from those individuals. Considering team combinations in this way provides a better estimate of condition characteristics than simply randomly assigning individuals to teams [157].

5.5.2 Materials

Participants were provided with a design statement that laid out a design scenario and the requirements for their solution. This design statement is shown in Figure 5.2. Participants were instructed to design their system in order to minimize the peak temperature in the home (preferably below 24°C) and minimize the total cost of the system (preferably below $20,000). Further, they were permitted a total of 50 design actions each over the course of a 30-minute design session. One design action was defined as the addition or removal of a product, the relocation of a product within the home, or the modification of the parameters of a cooler (flow rate or power).

Design Context
Alex recently purchased an old house in Arizona that doesn't have central air conditioning. The floor plan is shown below. Instead of retrofitting the house with a new central system, Alex wants to use connected products to keep the house cool during the summer.

Design Objectives
Your team has been hired to design a system of connected products to control the temperature in Alex's home. Your team's objectives are the following:

1. Minimize the peak temperature within the home (preferably below 24°C)
2. Minimize the total cost of the system (preferably below $20,000). Remember, this includes the component cost as well as the operating cost.

Constraints

- Total time: 30 minutes
- Total actions: 50 per team member
- ***You must complete all design actions within the allotted 30 minutes.***

FIGURE 5.2. DESIGN STATEMENT FOR COOLING SYSTEM DESIGN TASK.

To facilitate the design process, each participant was given access to a computer on which was loaded a design interface (see Figure 5.3). This interface served a number of critical functions. First, the design interface allowed participants to construct and evaluate solutions and provided immediate feedback on design quality after every design modification. It also tracked every operation performed by the participants, enabling the reconstruction of each team's search for solutions following the conclusion of the study. Finally, the design interface prompted teams to interact at the correct intervals and indicated how many actions they had left during the study. Teams in Condition 1 did not interact, teams in Condition 2 interacted after every 10 actions,, and teams in Condition 3 interactions after every 5 actions.

FIGURE 5.3. DESIGN INTERFACE FOR COOLING SYSTEM DESIGN TASK.

5.5.3 Procedure

The study took 55 minutes in total. The allocation of time during the experiment is shown in Figure 5.4. After giving informed consent and having an opportunity to ask any preliminary questions, participants completed a pre-survey that collected basic demographic information. Participants were then given a handout that, together with a tutorial program, guided them through the functionality of the design interface. The tutorial program was identical to the design interface in every way except for the use of a simplified residential structure layout and the lack of a limit on the number of operations. Following the tutorial, each participant was given the design statement shown in Figure 5.2 and given three minutes to read it. Participants were then allowed to open the design interface and start solving the design problem. During the 30 minutes allotted for design, every participant was instructed to complete 50 total design actions. Participants in Condition 1 worked individually and uninterrupted. Participants in Conditions 2 and 3 were prompted by the design interface to interact with their team at regular intervals (after every 10 individual actions for teams in Condition 2, and after every 5 for those in Condition 3).

Participants were only permitted to speak within their team and share their current solutions while interacting. However, they were not allowed to make any modifications to their current solutions during interaction periods, in order to ensure that their focus was on interacting with the teammates rather than continuing to work on their own solution. Once participants decided to cease interacting and continue working they were required to work individually until they were prompted to interact again. Once the 30-minute design session ended, participants completed a post-survey. The post-survey was intended to assess participants' perception of their design activities and their perception of the design problem itself.

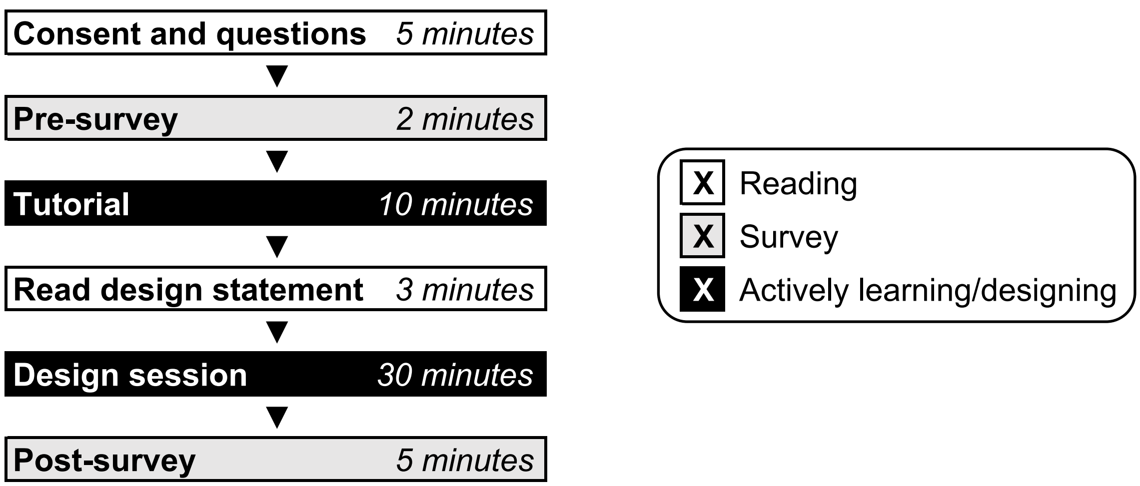


FIGURE 5.4. TIME ALLOCATION DURING COOLING SYSTEM DESIGN STUDY.

5.6 Results

The results from the cognitive study are presented in two sections. In the first section we present results from the cognitive study to examine the performance achieved by teams in each of the three conditions. Initially, little difference between the conditions is observed, due to the non-normality of the distribution of solutions. Following the application of a log-transformation to increase normality, stronger trends are observed that support the prediction made in previous sections of the chapter. The second section presents trends observed in the post-survey that queried perceived difficulty, personal satisfaction, and preferred interaction.

5.6.1 Performance Analysis

A plot of the total cost and peak temperature of each team's best solution is provided in Figure 5.5. A visual examination of this plot indicates that the distribution of the data is non-normal. The non-normality is particularly exacerbated by several extreme solutions with total costs below $19000 and a single solution with peak temperature greater than 26°C. An Anderson-Darling test for normality corroborates this observation for both peak temperature ($A^2 = 1.103, p < 0.01$) and total cost ($A^2 = 2.321, p < 0.01$).

In order to correct this non-normality a log-transformation was applied to each of the variables [158]. Figure 5.6 plots the best solutions from each team after this transformation. The extreme data points are no longer apparent, and the distribution appears much more normal. The visual normality of the data is confirmed by the Anderson-Darling normality test for both peak temperature ($A^2 = 0.439, p > 0.1$) and total cost ($A^2 = 0.514, p > 0.1$).

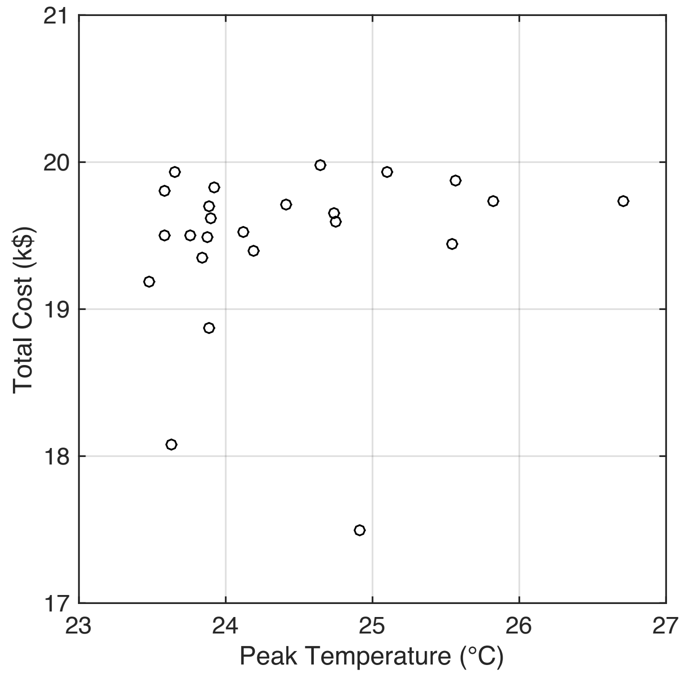


FIGURE 5.5. PLOT OF PEAK TEMPERATURE VERSUS TOTAL COST FOR BEST SOLUTIONS FROM EACH TEAM.

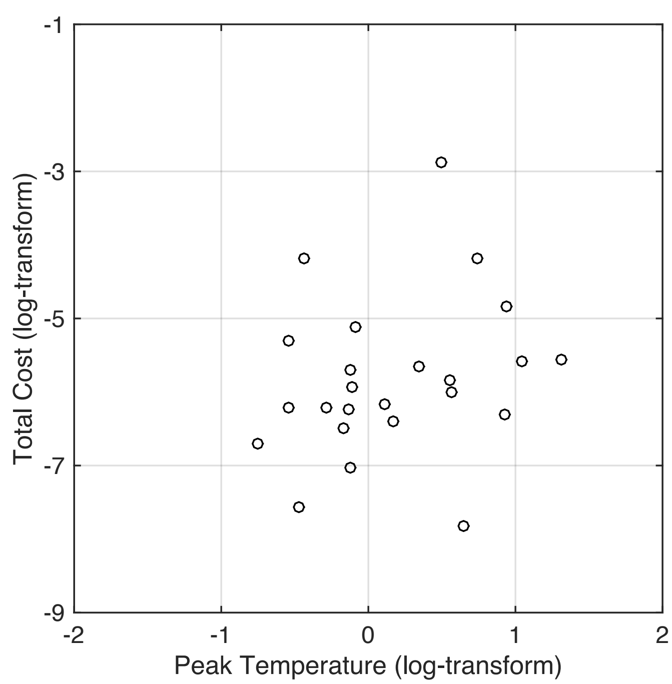


FIGURE 5.6. PLOT OF PEAK TEMPERATURE VERSUS TOTAL COST FOR BEST SOLUTIONS FROM EACH TEAM AFTER LOG TRANSFORMATION.

Figure 5.7 shows the mean quality of the best solutions achieved by teams in each of the conditions. Figure 5.7.a shows total cost (measured in thousands of dollars), and Figure 5.7.b shows peak temperature (measured in degrees Celsius). The mean for each condition is shown with a square marker and error bars that indicate ±1 standard error. Teams in Condition 1 (no interaction) achieved solutions with significantly lower cost than teams in Condition 3 (0.2 interaction frequency). Pairwise comparisons between Condition 1 and Condition 2, and between Condition 2 and Condition 3 were not significant. There was also a statistically significant correlation between interaction frequency and total cost ($F = 17.6, p < 0.001$). No significant between-condition comparisons or trends were discovered with respect to peak temperature. This may indicate that participants were more comfortable minimizing for cost and preferred to treat peak temperature more like a constraint.

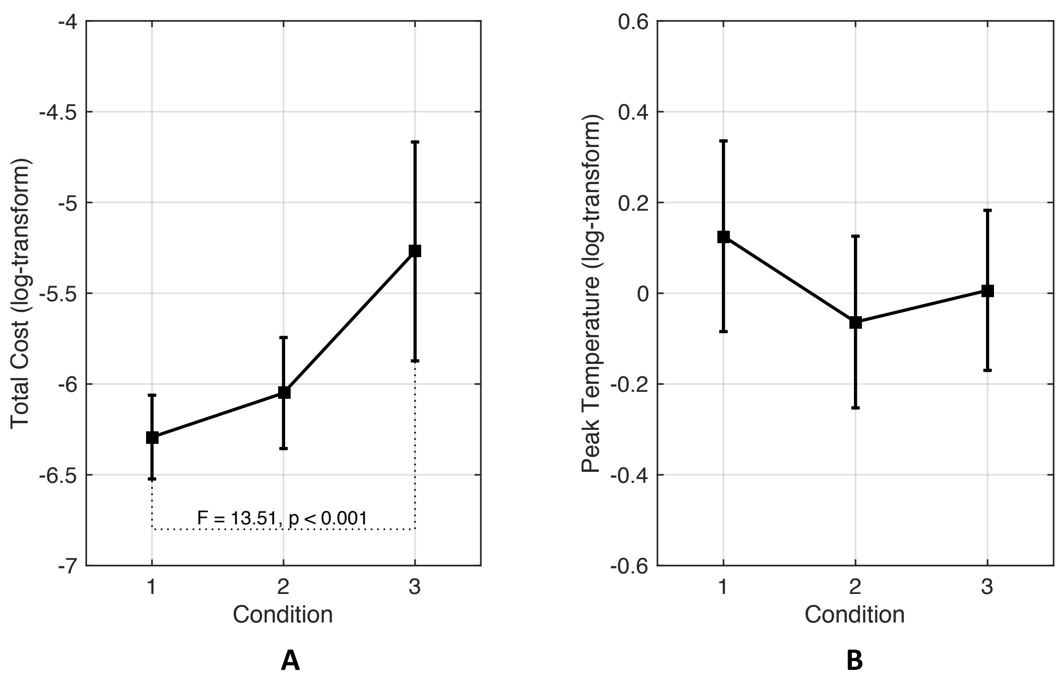


FIGURE 5.7. QUALITY OF BEST SOLUTIONS WITH RESPECT TO LOG-TRANSFORMATIONS OF (A) TOTAL COST, AND (B) PEAK TEMPERATURE. ERROR BARS SHOW ±1 S.E.

This analysis indicates that teams using lower interaction frequencies tended to produce better solutions. This result is consistent with the prediction that non-interacting teams (Condition 1) would achieve the best solutions to the cooling system design problem, and thus provides a limited validation for the computationally-derived predictive equations. This is a surprising result, given the common assumption that teams are better than individuals.

It is usually assumed that teams should outperform individuals on a given task. One phenomenon that has been cited to support this assumption is social facilitation [149]. Social facilitation is the tendency of individuals to perform better when in the presence of others, especially for tasks that have been rehearsed [159,160]. All participants in this study practiced the actions involved in solving the problem during the tutorial, and participants in Conditions 2 and 3 accomplished the design task in the presence of others (their team). A post-study survey also revealed that individuals who interacted more frequently were significantly more satisfied with their performance, indicating that they were affected by interaction with their team. Therefore, social facilitation should have been present in Conditions 2 and 3 (boosting their performance) but not for Condition 1 (in which participants worked individually). Despite this expected motivational effect, Condition 1 still displayed mean performance that was on better on average than the interacting team-based conditions. This underscores the influence that the properties of a design problem can exert on the effectiveness of the problem-solving approaches used by designers. However, this is not a blanket finding – there may (and probably do) exist problems for which social facilitation may alter the balance in favor of interacting teams, or for which interacting teams perform better outright. Therefore, in considering the selection of an optimal interaction frequency it is important to balance the potential benefit from social facilitation against the values of problem properties.

5.6.2 Survey Results

The previous section addressed the quality of the best solutions achieved by teams in each condition. This section addresses the participants' responses to a survey that they completed following the design portion of the study. The questions in the survey were designed to query participants' perceptions of the design task and opinions regarding their team's actions.

Participants were first asked to rate how difficult they perceived the design problem to be on a scale from 1 (very easy) to 7 (very difficult). There was a moderately significant positive correlation between interaction frequency and perceived difficulty of the design task ($F = 3.10$, $p < 0.10$). Participants in Condition 1 reported significantly lower difficulty than participants in Condition 3 ($F = 3.34$, $p < 0.10$). Pairwise comparisons between Condition 1 and Condition 2, and between Condition 2 and Condition 3 were not significant. These results are shown graphically in Figure 5.8.

This shows that teams that solved the problem with higher interaction frequencies were likely to report higher perceived task difficulty. It is likely that this was due to the simple fact that teams that interacted less frequently tended to achieve solutions with higher quality (see Figure 5.7) and the lower perceived difficulty is simply an indication that they solved the problem with greater ease. Another explanation is that teams in conditions 2 and 3 had an additional burden associated with interacting, which led to the perceived increase in the difficulty of the design problem.

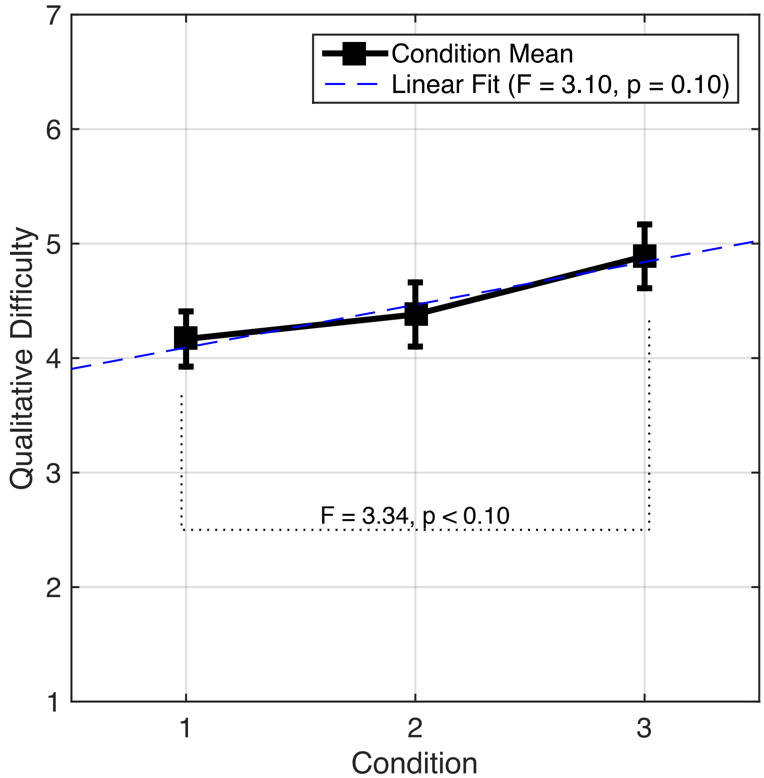


FIGURE 5.8. PERCEIVED DIFFICULTY OF THE COOLING SYSTEM DESIGN TASK AS REPORTED IN THE POST-SURVEY.

Participants were also asked how satisfied they were with their personal performance in the design task from 1 (very dissatisfied) to 7 (very satisfied). There was a significant positive correlation between interaction frequency and personal satisfaction with performance on the design task ($F = 4.17, p < 0.05$). Further, participants in Conditions 1 reported significantly lower satisfaction than participants in Condition 3 ($F = 5.69, p < 0.05$). Pairwise comparisons between Condition 1 and Condition 2, and between Condition 2 and Condition 3 were not significant. These results are shown graphically in Figure 5.9.

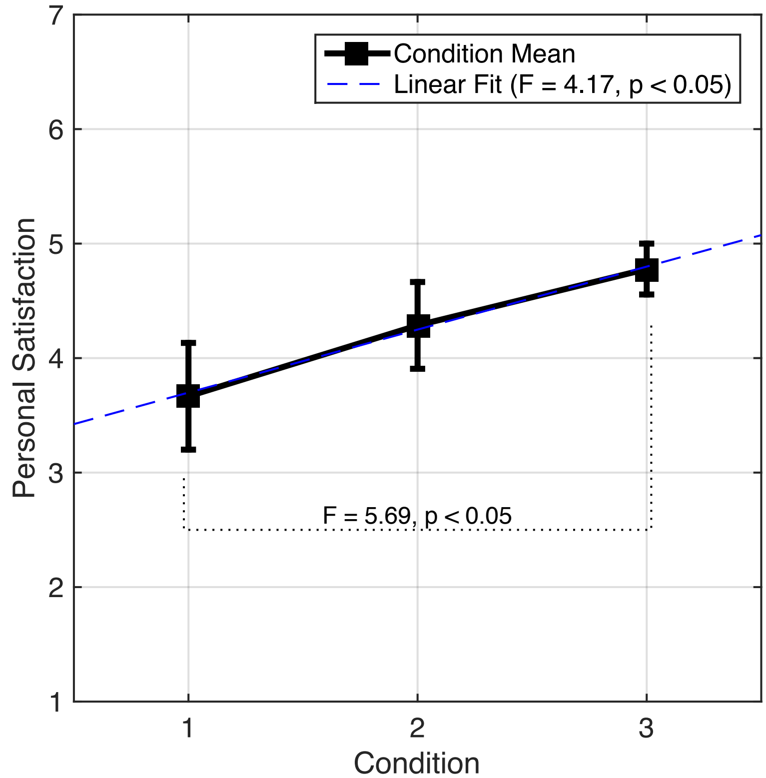


FIGURE 5.9. SATISFACTION WITH PERSONAL PERFORMANCE IN THE COOLING SYSTEM DESIGN TASK AS REPORTED IN THE POST-SURVEY.

Teams with higher interaction frequencies experienced greater difficulty while solving the problem (see Figure 5.8). One might expect that this would lead participants in frequently interacting teams to experience less satisfaction with their individual performance. However, the opposite is true – higher interaction led to higher personal satisfaction. A key dimension of human happiness is derived from relatedness and belonging [161], needs that can be fulfilled by interaction within a team. Therefore, it is not surprising that participants in more frequently interacting teams were more satisfied. More frequent interaction can also develop greater cognitive consensus [162], which has been correlated with greater satisfaction [163]. This offers another plausible explanation for the correlation between interaction frequency and personal satisfaction from the cognitive study. It is indicative of the strength of these social effects that they reversed any negative impact on satisfaction that might have resulted from lower solution quality.

Participants in the team-based conditions (Conditions 2 and 3) were also asked to indicate whether they would have liked to interact with their team less often, equally as often as they did, or more often. A summary of responses is shown in Figure 5.10. If participants had some intuition about how to improve their performance by changing interaction frequency, it would be expected that participants in Condition 3 (being furthest from the optimal interaction frequency) would have a stronger preference for lower interaction than participants in Condition 3. However, a χ^2-goodness-of-fit test indicates that there is not a significant difference between the response frequencies of the two conditions ($\chi^2 = 2.46, p > 0.20$).

The lack of a significant difference between the interaction preferences of the two conditions indicates that participants were unable to identify the appropriate interaction level that would be suitable for the task. This means that design teams may lack the ability to internally regulate interaction, and thus need to rely on external tools like those introduced in Chapter 4 and validated in the current chapter to select an optimal interaction frequency. In both conditions, a plurality of the participants preferred no chance in their frequency of interaction. However, of those who preferred a change, more preferred to interact more frequently than less frequently. This trend is apparent in both conditions. This demonstrates that individuals' preferences for interaction frequency may in fact lead them to sub-optimal behavior.

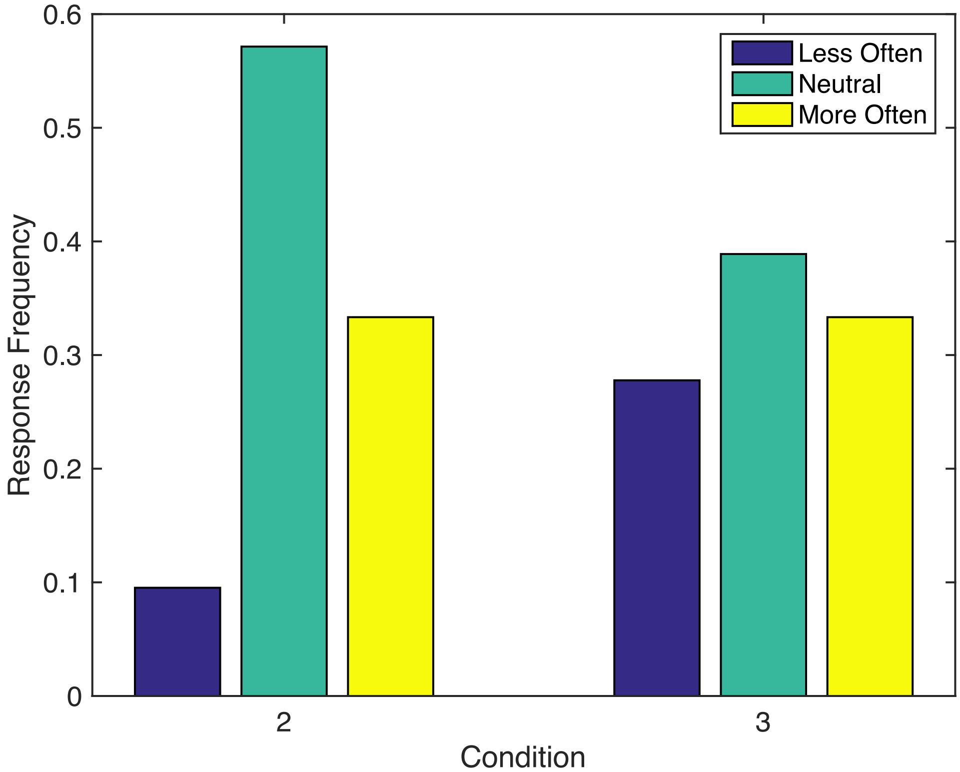


FIGURE 5.10. DESIGN INTERACTION LEVEL FOR THE COOLING SYSTEM DESIGN TASK AS REPORTED IN THE POST-SURVEY.

The above analysis assumes that participants' desire to interact is performance driven. While higher interaction led to worse performance, it also led to a higher degree of personal satisfaction (a trend demonstrated in Figure 5.9). Therefore, it may be that participants' preferences for more frequent interaction were driven by a desire for personal satisfaction rather than any intuition about how to achieve better performance. This preference for more frequent interaction may also be evidence of a type of optimism bias, which is the tendency for individuals to believe they are less at risk of a negative outcome than their peers [164]. Participants desired to obtain higher satisfaction by interacting more frequently and believed (falsely and optimistically) that they would be capable of maintaining their current level of performance in doing so.

5.7 Summary

Chapter 4 presented a procedure for measuring the properties of a design problem and developed relationships for predicting optimal team characteristics for solving that problem. This chapter conducted a cognitive study to experimentally offer a limited validation of those relationships. A design problem was first defined involving the creation of a cooling system for a home. Then, the properties of that design problem were measured, and the values of those properties were used to predict the optimal interaction frequency to be used by a team when solving the problem. Finally, this prediction was tested in a cognitive study in which participants solved the cooling system design problem in several different conditions. It was predicted that non-interacting teams would perform the best on the design task, or at least that lower interaction levels would be heavily preferred. The results from the cognitive study indicated that teams with no interaction and low interaction had similar performance levels, which were significantly higher than the performance of teams with higher interaction. This provided a limited validation of the prediction. The task used here assumed that all individuals had similar skillsets. For design problems that require diverse skillsets or perspectives it is less likely that zero interaction between team members will be the preferred approach.

In addition to validating the methodology and relationships developed in the previous chapter, the work in this chapter investigated the impact of interaction frequency on perceived problem difficulty, personal satisfaction, and preferred interaction frequency. It was found that higher interaction led to high personal satisfaction as well as higher perceived task difficulty. There was also no significant difference between the preferred interaction frequencies reported by participants in the two team-based conditions. Therefore, it is unlikely that members of a team possess accurate self-regulating mechanisms or intuitions that direct them towards optimal interaction patterns, and may have a tendency to interact more often than necessary because of a desire for greater personal satisfaction that comes with working closely with teammates.

Future work should perform more comprehensive validation of the predictive procedure used here using a large and diverse set of design problems. This would also provide the opportunity to refine the predictive procedure. However, research should also investigate the social cost of optimal interaction, specifically by investigating the correlation of interaction frequency with personal satisfaction in longitudinal design tasks. This may have important implications for long-term maintenance of morale in teams and organizations.

Chapter 6: The Heterogeneous Simulated Annealing Teams Optimization Algorithm[4]

A thing is never only itself.
Sometimes it rhymes.

Dean Young

6.1 Overview

Focused research has uncovered mechanisms of design cognition and revealed insights that can be used to inform the methods used by human designers [42,165,166]. A number of studies have also sought to inform better computational optimization and design tools from the results of design studies. These include the use of machine learning algorithms to algorithmically encode stylistic aspects of design [167], design tools that draw upon empirical studies of human analogy use [168], and research aimed at enhancing the synergy between humans and computational agents [169].

This chapter uses the CISAT modeling framework (introduced in Chapter 3) to construct a novel optimization algorithm that emulates aspects of human design teams. The underlying structure of CISAT is based on simulated annealing (SA) [35], a stochastic optimization algorithm that has seen extensive use in engineering design and other fields [87,90,170–174]..

[4] This chapter is based on:

McComb, C., Cagan, J., and Kotovsky, K., 2015, "Drawing Inspiration From Human Design Teams for Better Search and Optimization: The Heterogeneous Simulated Annealing Teams Algorithm," Journal of Mechanical Design, **138**(4), pp. 044501-1 - 044501-6.

Because of this relationship, the question arose: *Can aspects of human problem-solving be used to inform a new variation of SA via CISAT?* By selectively retaining several CISAT characteristics, this chapter creates an SA-based numerical optimization algorithm that incorporates beneficial aspects of engineering design teams.

Interaction between members of a team enables them to divergently explore a design space, and later convergently focus their efforts on a diminishing set of alternatives [44]. In order to accomplish the divergent stage, members of a team must be capable of independently tailoring their approach as they search for solutions – a behavior accounted for by the *locally-sensitive search* characteristic in CISAT. This specifically reflects the ability of expert designers to use a mixture of depth- and breadth-first solution strategies [69]. To accomplish the convergent stage, the members of the team must have some mechanism for interacting and sharing solutions – a behavior accounted for by the *quality-informed solution sharing* characteristic of CISAT. This reflects the fact that members of a design team factor design quality into decisions, but are also able to pursue designs that may currently display lower quality [36].

This chapter introduces the Heterogeneous Simulated Annealing Team (HSAT) algorithm. This algorithm is based on the CISAT framework, and specifically retains the two CISAT characteristics indicated above: locally-sensitive search, and quality-informed solution sharing. The HSAT algorithm is compared to a variety of SA-based algorithms on three benchmarking functions, and consistently provides terminal solutions that are better on average than other SA-based algorithms explored.

6.2 Background

The conventional SA algorithm is based on the physical annealing process, in which materials are heated and cooled in a controlled manner to minimize residual stresses [35]. When the material is hot, the movement of atoms is random in nature; atoms may even move occasionally in a direction that *increases* potential energy. As the material is cooled, atomic movements become more deterministic and atoms begin to move almost entirely in directions that decrease potential energy.

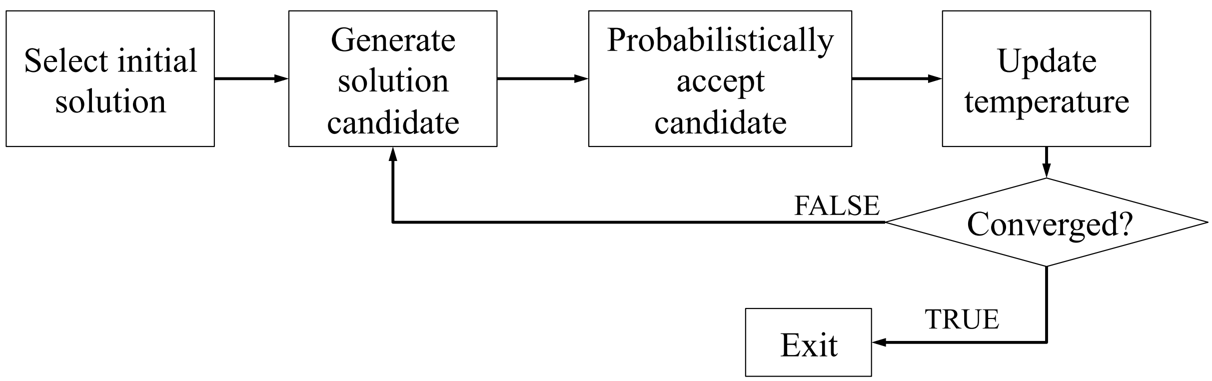


FIGURE 6.1. FLOWCHART FOR CONVENTIONAL SIMULATED ANNEALING ALGORITHM.

In the analogous optimization algorithm, the goal is to minimize the value of an objective function, rather than potential energy. A conceptual flowchart of the conventional SA algorithm is provided in Figure 6.1. With the initialization of the algorithm, an initial solution candidate is generated (either at random or using some sort of heuristic). The algorithm then iterates to improve the solution. Within each iteration, a new solution candidate is generated. It is then probabilistically accepted or rejected according to its objective function value and the temperature of the annealing schedule. The temperature is then updated, and the algorithm continues to the next iteration if convergence criteria have not been met. The probability of accepting a worse solution is typically decreased after every iteration, enabling the algorithm to progressively transition from initial stochastic search to final deterministic search. This procedure is described in greater mathematical detail below.

For the purposes of this chapter, initial solutions are selected uniformly at random within a continuous space that is defined by some upper and lower bounds. The precise values of those bounds are specific to the problem being solved. A new solution candidate, $\boldsymbol{x}^{\mathrm{NEW}}$, is created by drawing a vector at random from the Cauchy distribution and adding the resulting vector to the current solution, $\boldsymbol{x}_i$. This is accomplished by computing

$$\boldsymbol{x}^{NEW} = \boldsymbol{x}_i + T_i \cdot tan\left(uniform\left(-\frac{\pi}{2}, \frac{\pi}{2}, D\right)\right), \qquad 6.1$$

where T_i is the current temperature. The function uniform draws a point at random from the continuous D-dimensional space with an upper bound of $\frac{\pi}{2}$ and a lower bound of $-\frac{\pi}{2}$ in each direction. The Cauchy distribution is preferable to the Gaussian distribution for this application because it has thicker tails, and thus encourages more extensive search [175]. Once a new solution is generated, the objective function is evaluated for that solution. If the solution is better than the previous solution (i.e., $f(\boldsymbol{x}_{i+1}) < f(\boldsymbol{x}_i)$) then it is accepted (i.e., $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}^{\mathrm{NEW}}$). If the new solution is *not* better than the previous solution, it is still accepted with some probability p, which is defined as:

$$p = exp\left(\frac{f(x_{i+1}) - f(x_i)}{T_i}\right), \qquad 6.2$$

If the new solution is not accepted, then the previous solution is carried into the next iteration (i.e., $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i$).

An important consideration in any SA algorithm is the annealing schedule, which dictates how the simulated temperature (T_i) changes over time. The temperature dictates the probability of accepting a worse solution, and can in this way have an effect on the progressive transition from exploration of the problem space to exploitation. The temperature may also influence the generation of new solution candidates in some SA algorithms. The most rudimentary annealing schedules are monotonically decreasing functions of iteration number. Theoretically, the probability that the algorithm will find the global optimum approaches unity as the annealing schedule is stretched over an increasing number of iterations [176]. However, computational resources are finite, which makes the use of extremely long annealing schedules impractical. Therefore, the implementation of adaptive annealing schedules has been of particular interest to simulated annealing practitioners. For instance, a schedule proposed by Azizi and Zolfaghari follows a conventional geometric annealing schedule, but increases the temperature for every consecutive uphill move that is made [177]. The intuition is that this will help the algorithm to escape local minima. This adaptive schedule was found to perform better than conventional annealing schedules on a job-scheduling task. Triki et al. demonstrated that many common adaptive temperature schedules follow very similar forms, and identified a single term within the temperature update rule that differentiated between schedules [91]. While other schedules defined this term using a function of objective function variance, or number of accepted solutions, Triki et al. proposed a new schedule that simply replaced the term with a constant. This schedule performed better than another widely-used adaptive schedule [178] on a variety of benchmarking tasks.

Two annealing schedules are used in this chapter: the classical Cauchy schedule and the Triki adaptive schedule. The Triki adaptive annealing schedule is an integral part of the HSAT algorithm, while the Cauchy annealing schedule is used exclusively for comparison. The temperature is not updated after every iteration, but instead updated intermittently every n iterations (i.e. dwell time). For the Cauchy schedule, the temperature is updated as:

$$T_{i+1} = \frac{T_0}{1 + \delta_C \cdot i}, \quad 6.3$$

Where T_0 is the initial temperature, i is the index of the current iteration, and δ_C is a parameter that allows the schedule to be extended or compressed. The second schedule (referred to as the Triki schedule for the remainder of this chapter) updates temperature as

$$T_{i+1} = T_i\left(1 - \frac{T_i \cdot \delta_T}{\sigma_{f(x)}^2}\right), \quad 6.4$$

where δ_T is a parameter that controls how quickly adaptation occurs, and $\sigma_{f(x)}^2$ is the variance of the objective function value for candidate solutions evaluated since the last temperature update.

A number of algorithms use SA constructs as parallel sub-routines, and combine solutions at regular intervals using computational genetic operators [172,179]. Recent work has also developed Multi-agent Simulated Annealing (MSA) algorithms, which employ software agents to operate on multiple solutions. A software agent, usually referred to simply as an agent in this context, is a computational sub-routine that operates on potential solutions. The MSA algorithms utilize principles of differential evolution and particle swarm optimization to accomplish interaction between agents [180,181]. However, the agents in those algorithms do not possess any sort of individual strategy or preference for exploring solutions. The ownership of personal characteristics is a common factor in many agent-based algorithms [86]. Personal characteristics can also lead to heterogeneity and diversity of agents, which can improve performance in agent-based optimization models [182]. Creating diversity between agents in an SA-based algorithm may similarly improve performance.

It should be noted that there are many alternatives to SA-based algorithms for global optimization. These include parallel Genetic Algorithms which utilize parallelism similar to some MSAs [183], basin-hopping algorithms [184], particle swarm optimization [185], efficient global optimization [186], branch-and-bound methods [187], and pattern search methods [188].

6.3 Heterogeneous Simulated Annealing Teams Algorithm

HSAT is a multi-agent simulated annealing algorithm that retains two characteristics from the CISAT modeling framework on which it is based: *quality-informed solution sharing* and *locally-sensitive search*. The interaction between agents in HSAT is structured according to the *quality-informed solution sharing* characteristic from CISAT. It should be noted that the retention of this characteristic also requires the *multi-agency* CISAT characteristic to be active. Since these two characteristics are convolved in this way, they are treated as a single characteristic in this chapter. The HSAT agents are also provided with individually-controlled adaptive temperature schedules, thus implementing the *locally-sensitive search* characteristic from CISAT. The overall structure of HSAT is identical to that of CISAT (see Figure 2.1).

When agents are instantiated, each selects a candidate solution at random from the design space. The HSAT algorithm then begins iterations to optimize the objective function. At the beginning of every iteration, the objective function of every agent's current solution is shared with the other agents in the team through the vector $\boldsymbol{F}$ such that

$$\boldsymbol{F} = [f(x_i^1), f(x_i^2), \ldots, f(x_i^N)]\,, \tag{6.5}$$

where N is the total number of agents. Note that subscripts indicate iteration number, while superscripts indicate different agents. A new vector $\boldsymbol{W}$ is then defined as the relative function value of each current solution compared against the worst current solution

$$\boldsymbol{W} = -\boldsymbol{F} + max(\boldsymbol{F}). \tag{6.6}$$

Note that this formulation of the weighting vector applies only to minimization problems such as those used in this chapter. Conversion to a maximization problem can be accomplished by changing the size of the objective function and taking the minimum value of the vector $\boldsymbol{F}$ instead of the maximum.

Each agent handles the remaining operations in the iteration independently by first select a starting solution using the equation

$$j = mult\left(\frac{W}{\sum_i W_i}\right), \tag{6.7}$$

where the function mult returns a draw from a multinomial distribution defined by the vector of probabilities proportional to the weighting vector $\boldsymbol{W}$. This quality-informed solution-sharing procedure probabilistically encourages agents to pursue the best solutions. The solution selected through this process, $\boldsymbol{x}_i^j$, is then used by the agent to begin the next iteration. The equation for calculating the new solution candidate for agent k is therefore

$$x^{k,NEW} = x_i^j + T_i^k \tan\left(uniform\left(\frac{\pi}{2}, \frac{\pi}{2}, D\right)\right). \tag{6.8}$$

In other words, the agent begins at the starting solution $\boldsymbol{x}_i^j$ (selected with Equation 6.7) and then applies a Cauchy modification to it, similarly to the conventional SA algorithm. If the new solution candidate, $x^{k,\mathrm{NEW}}$, is better than the agent's previous solution, x_i^k, then the solution candidate is accepted. If it is not better, then the agent still accepts the solution with the acceptance probability computed with Equation 6.2. Finally, the temperature is updated using the Triki annealing schedule (see Equation 6.4). It should be noted that temperature is updated independently by each agent, allowing agents to develop heterogeneous strategies.

6.4 Comparison Methodology

6.4.1 Compared Algorithms

The HSAT algorithm employs two features inspired by characteristics observed in human design teams. In order to fully understand the impact of these features, HSAT is compared to three other SA-based algorithms: Non-Adaptive SA, Adaptive SA, and Non-Adaptive MSA. Non-Adaptive SA incorporates neither of the unique features of HSAT. It employs a classical annealing schedule, and only a single agent. Adaptive SA still employs only a single agent, but uses an adaptive annealing schedule, thus incorporating one of the HSAT features. Non-Adaptive MSA incorporates only the other HSAT feature, employing multiple agents, yet using only a single non-adaptive annealing schedule. A summary of all four SA-based algorithms is provided in Table 6.1.

TABLE 6.1. SUMMARY OF SA-BASED ALGORITHMS COMPARED TO THE HSAT ALGORITHM.

Algorithm	**Multiple agents?**	**Individual adaptivity?**	**Description**
Non-Adaptive SA	No	No	Single-agent simulated annealing algorithm using Cauchy annealing schedule
Non-Adaptive MSA	Yes	No	Multi-agent simulated annealing algorithm with quality-informed interaction between agents, but with a classical annealing schedule.
Adaptive SA	No	Yes	Single-agent simulated annealing algorithm using Triki annealing schedule
HSAT	Yes	Yes	Multi-agent simulated annealing algorithms with quality-informed interaction between agents, and a separate adaptive annealing schedule for each agent

Because SA algorithms progressively transition from stochastic to deterministic search, a purely stochastic algorithm (random search) and a purely deterministic algorithm (gradient-based) are used for comparison. The random search algorithm samples randomly within the bounds of the search space at each iteration, and returns the best solution encountered. The gradient-based algorithm is the Broyden–Fletcher–Goldfarb–Shanno interior-point algorithm, a gradient-based algorithm that approximates the Newton method. In the remainder of the chapter, this algorithm will be referred to as the *gradient algorithm*. Rather than directly computing the Hessian, this algorithm estimates it using successive gradient information. Because it avoids the costly computation of the Hessian matrix, it is ideally suited to high-dimensional spaces. A more detailed description of the gradient algorithm can be found in [189]. Since genetic algorithms are typically used to optimize multi-modal functions, a comparison is also made against the standard genetic algorithm provided in MATLAB [190,191].

For equivalent comparison, every algorithm (both SA-based, and non-SA-based) is permitted the same number of objective function evaluations during each run. The gradient algorithm is restarted with a new, random location every time it converges on a local minimum, until the allotted number of function evaluations is reached.

6.4.2 Benchmarking Functions

Algorithm performance is assessed with respect to three continuous functions. These functions are the Ackley function [192], the Griewank function [193], and the Rastrigin function [194]. The general forms of these functions and their bounds are provided in Equations 6.9 (the Ackley function), 6.10 (the Griewank function), and 6.11 (the Rastrigin function). The variable D indicates the number of dimensions in the search space.

$$f(\boldsymbol{x}) = -20\,exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{D} x_i}{D}}\right) - exp\left(\frac{\sum_{i=1}^{D} cos(2\pi x_i)}{D}\right) + 20 + exp(1) \tag{6.9}$$

$$-10 \leq x_i \leq 10 \quad \forall i$$

$$f(\boldsymbol{x}) = -\frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{6.10}$$

$$-600 \leq x_i \leq 600 \quad \forall i$$

$$f(\boldsymbol{x}) = \sum_{i=1}^{D} (x_i^2 - 10\,cos(2\pi x_i) + 10) \tag{6.11}$$

$$-5.12 \leq x_i \leq 5.12 \quad \forall i$$

The global minimum of all three functions is $f(\boldsymbol{x}^*) = 0$, and occurs at $\boldsymbol{x}^* = \boldsymbol{0}$. For the numerical experiments conducted in this chapter, every equation is implemented with 30 dimensions. For purposes of illustration, representations of these functions in two and three dimensions are provided in Figures 6.2, 6.3, and 6.4. The global minimum is shown in the two dimensional representation with a white star.

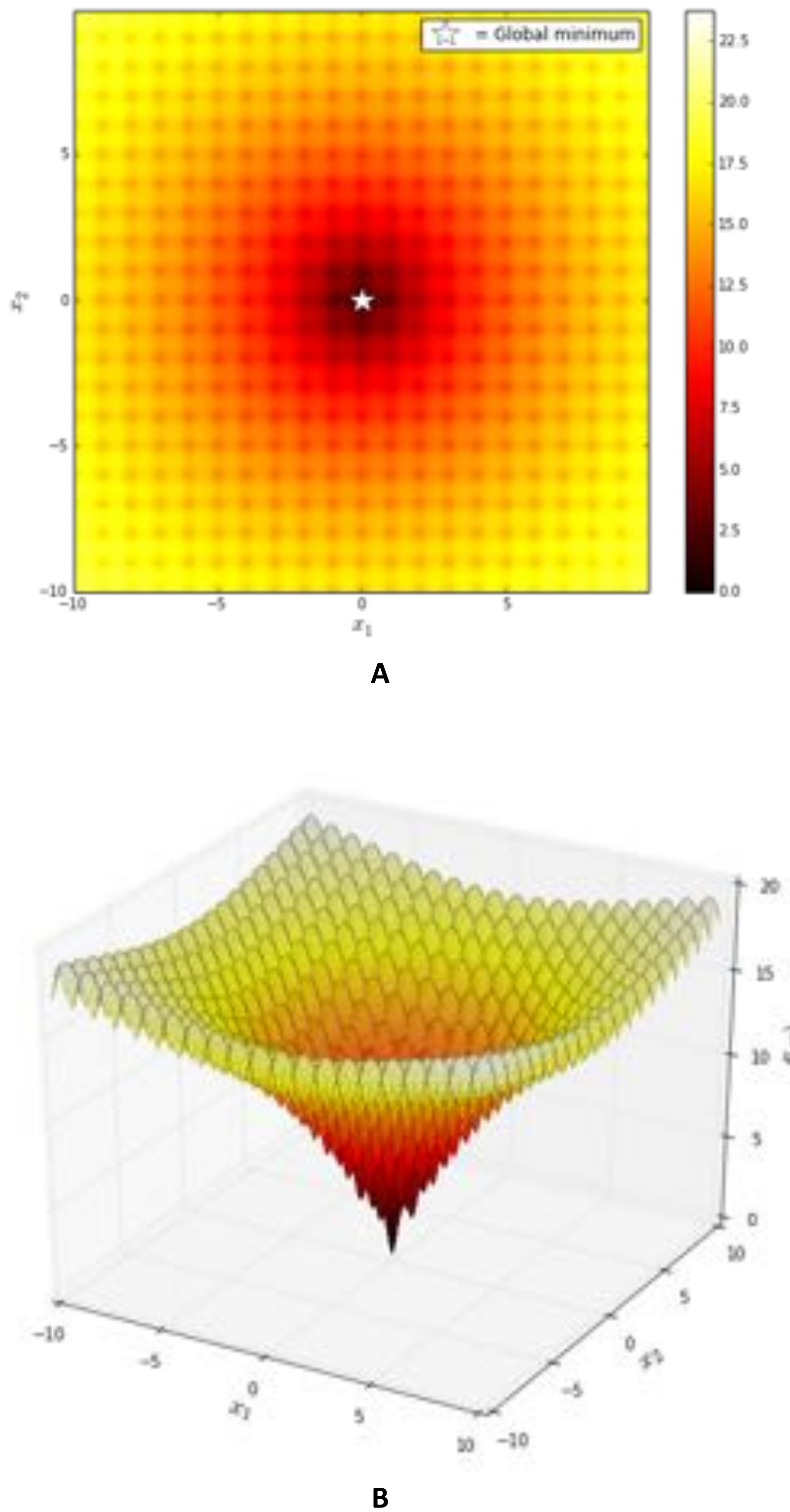


FIGURE 6.2. THE ACKLEY FUNCTION IN (A) TWO DIMENSIONS AND (B) THREE DIMENSIONS.

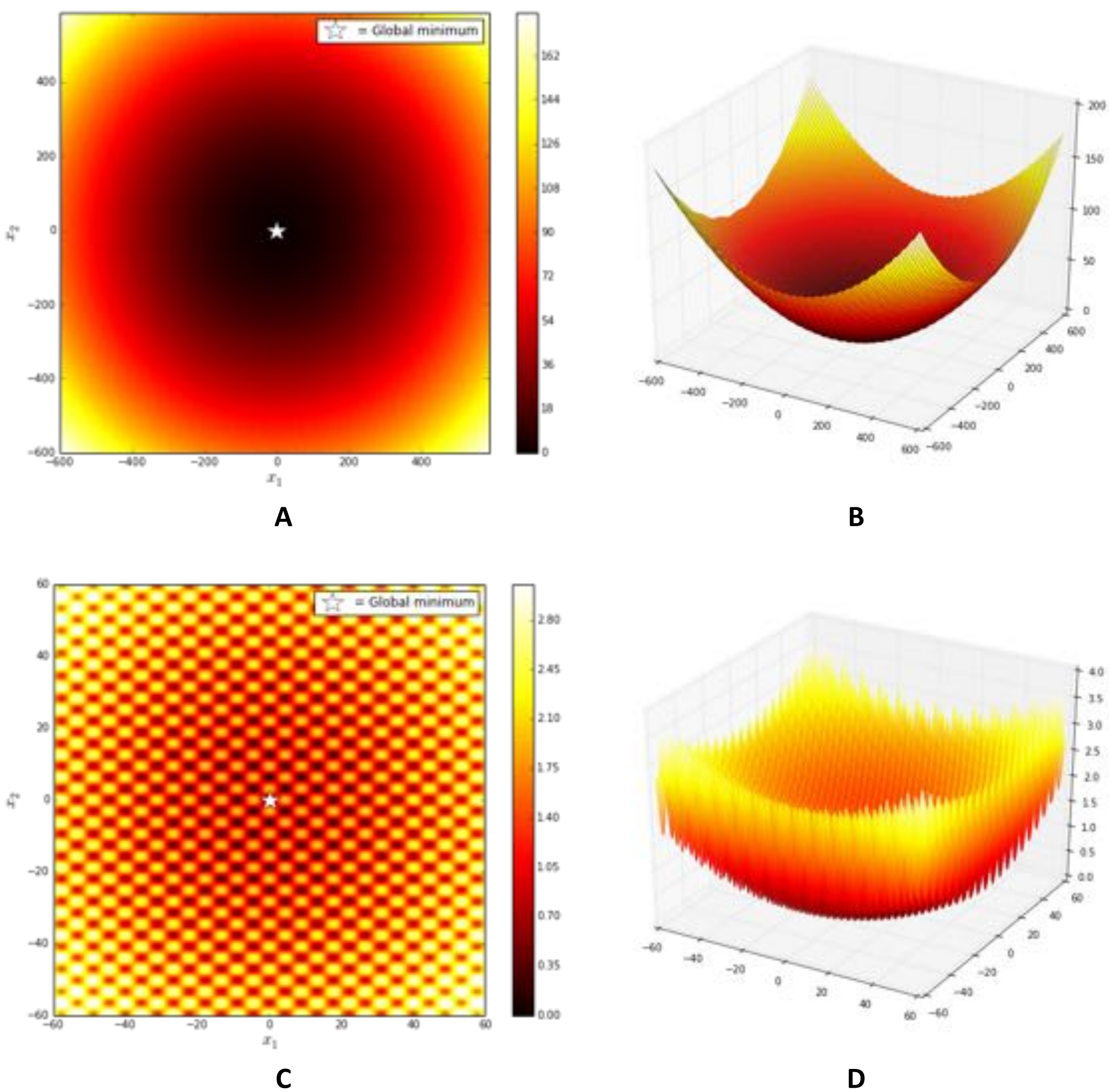
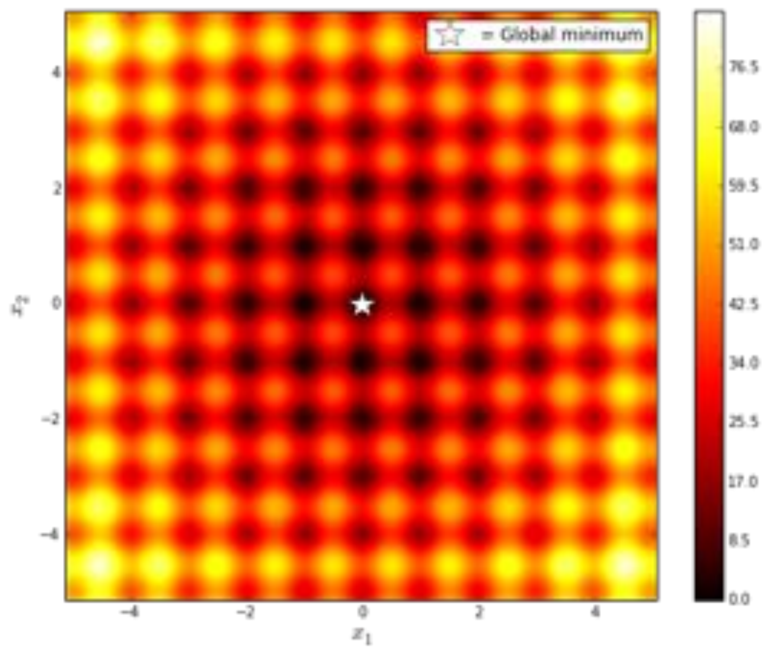
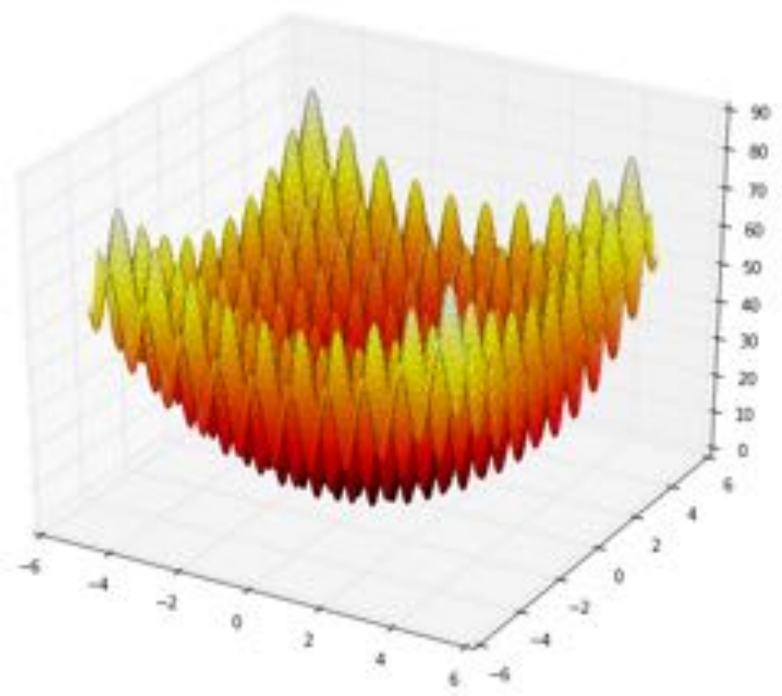


FIGURE 6.3. THE GRIEWANK FUNCTION SHOWING GLOBAL BEHAVIOR IN (A) TWO-DIMENSIONS AND (B) THREE-DIMENSIONS, AND LOCAL BEHAVIOR IN (C) TWO-DIMENSIONS AND (D) THREE-DIMENSIONS.



A



B

FIGURE 6.4. THE RASTRIGIN FUNCTION IN (A) TWO DIMENSIONS AND (B) THREE DIMENSIONS.

Each of these functions presents distinct challenges. The global minimum of the Ackley function resides in a central well, while much of the function is fairly flat. Therefore, minimizing this function requires a broad search to find the well, and then a local search to find the global minimum. In other words, an effective search requires breadth, followed by depth. The Griewank function is globally convex, but in the neighborhood of the global minimum there are many minima with very similar values. An effective minimization of this function would require depth (to follow the global behavior) followed by breadth (to search local minima). The Rastrigin function is composed of a number of deep wells, all of which contain local minima that are similar in value to the global minimum. Therefore, this function requires a combination of breadth (to search multiple wells) and depth (to efficiently minimize within each well).

A total of 100,000 function evaluations are allowed when algorithms solve the Ackley and Griewank functions. However, due to its challenging contour, algorithms are permitted 250,000 objective function evaluations when solving the Rastrigin function.

6.4.3 Algorithm Parameter Selection

The SA-based algorithms that are compared in this chapter all differ significantly, either in terms of annealing schedule, number of agents, or both. Consequently, each algorithm has a unique set of parameters that control its execution, including aspects such as the way in which temperature is updated and the generation of new solutions. Simply assuming that using the same parameter value for every algorithm (for instance, an initial temperature of 0.5) would ensure good performance for all algorithms is naïve. In fact, it may be the case that parameter values that are near-optimal for one algorithm may be detrimental to the performance of another. For this reason, a meta-optimization is performed to select the best parameter values for each of the SA-based algorithms. By repeating this procedure separately for every combination of SA-based algorithm and benchmarking function, it is ensured that algorithms are compared in terms of their best possible performance.

The meta-optimization procedure incrementally tunes parameters with a regressing univariate search to improve the average objective function value of the terminal solutions. Each of the SA parameters is updated in turn. The process of updating a single parameter is explained here, and illustrated in Figure 6.5. First, holding all other parameters constant, the update parameter is varied randomly near its original value. Every time the update parameter is varied, an SA optimization is performed with the new parameter value. The terminal solution from the SA optimization is logged along with the new parameter value. This procedure is repeated a large number of times. A quadratic function is then fit to the logged solution data, treating the update parameter as the independent variable and terminal objective function value as the dependent variable. The fitted function is minimized, returning an improved value for the update parameter.

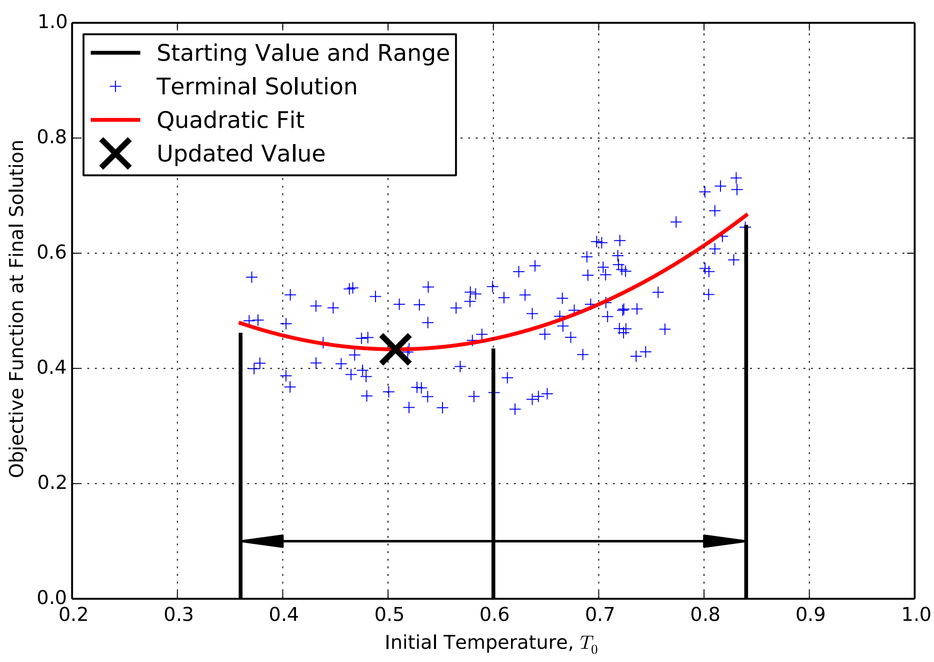


FIGURE 6.5. EXAMPLE SIMULATD ANNEALING PARAMETER UPDATE STEP USING REGRESSING UNIVARIATE SEARCH.

For the meta-optimization in this chapter, parameter values are varied within a range of ±40% of the current value. For each update, 100 different parameter values are evaluated. An example of a single parameter update using these values is shown in Figure 6.5. The starting value of initial temperature is 0.6, so the value is varied within the range of [0.36, 0.84]. A total of 100 SA optimizations are performed using different initial temperatures. A quadratic curve is fit to the terminal solution values and minimized. This yields an updated initial temperature of 0.515. Table 6.2 summarizes the parameters resulting from the application of the meta-optimization procedure to each combination of SA-based algorithm and benchmarking function.

TABLE 6.2. PARAMETERS USED FOR COMPARISON OF SA-BASED ALGORITHMS.

Algorithm	**Function**	**Number of Agents,** N	**Initial Temp.,** T_0	**Cauchy Param.,** δ_C	**Triki Param.,** δ_T	**Dwell Time,** n
	Ackley	1	5.77×10^{-1}	7.04×10^{-2}	-	3
Non-Adaptive SA	Griewank	1	7.14×10^{-1}	4.08×10^{-3}	-	10
	Rastrigin	1	3.92×10^{-1}	9.87×10^{-4}	-	11
	Ackley	1	4.17×10^{-1}	-	1.72×10^{-1}	126
Adaptive SA	Griewank	1	9.17×10^{-3}	-	4.12×10^{-7}	43
	Rastrigin	1	3.52×10^{-3}		6.50×10^{-1}	33
	Ackley	5	9.84×10^{-1}	4.34×10^{-1}	-	8
Non-Adaptive MSA	Griewank	8	1.52×10^{0}	2.12×10^{-2}	-	12
	Rastrigin	7	3.30×10^{-1}	8.87×10^{-3}	-	23
	Ackley	7	1.65×10^{-2}	-	2.91×10^{-1}	71
HSAT	Griewank	23	6.33×10^{-2}	-	7.47×10^{-1}	75
	Rastrigin	15	6.52×10^{-3}	-	1.97×10^{1}	31

6.5 Performance Benchmarking Results

Using the parameters from Table 6.2, each benchmarking function is solved 100 times with each of the algorithms. Cumulative distribution function for the objective function values of the terminal solutions are shown in Figures 6.6.a, 6.7.a, and 6.8.a. A cumulative distribution function indicates the probability that a random variable (in this case, objective function value) will have a value equal to or less than a given value on the x-axis of the plot. In other words, if a cumulative distribution function is given as $CDF(x)$, then $CDF(0.1)$ gives the probability that a single run of the algorithm will have a terminal objective function value equal to or less than 0.1. The cumulative distributions shown in Figures 6.6 - 6.8 are computed directly from the results of numerical experiments, and not smoothed in any way.

The best solution achieved by the algorithm over time is shown as a function of total objective function evaluations in Figures 6.6.b, 6.7.b, and 6.8.b. The geometric mean is used in these plots (rather than the arithmetic mean) because it better communicates central tendency when the data spans several orders of magnitude. The geometric mean of a vector Y is computed as $(\prod_{i=1}^{n} Y_i)^{\frac{1}{n}}$. Error bars are omitted from Figures 6.6.b, 6.7.b, and 6.8.b in the interest of visual clarity, but the spread of the terminal solutions can be inferred from the horizontal range of the cumulative distribution functions

Since parameters for SA-based algorithms are tuned for near-optimal performance for the given iteration limit, the average value of the best solution continues to improve throughout the allotted runtime (albeit slowly in some cases). Consequently, most SA-based algorithms achieve numerical convergence relatively late.

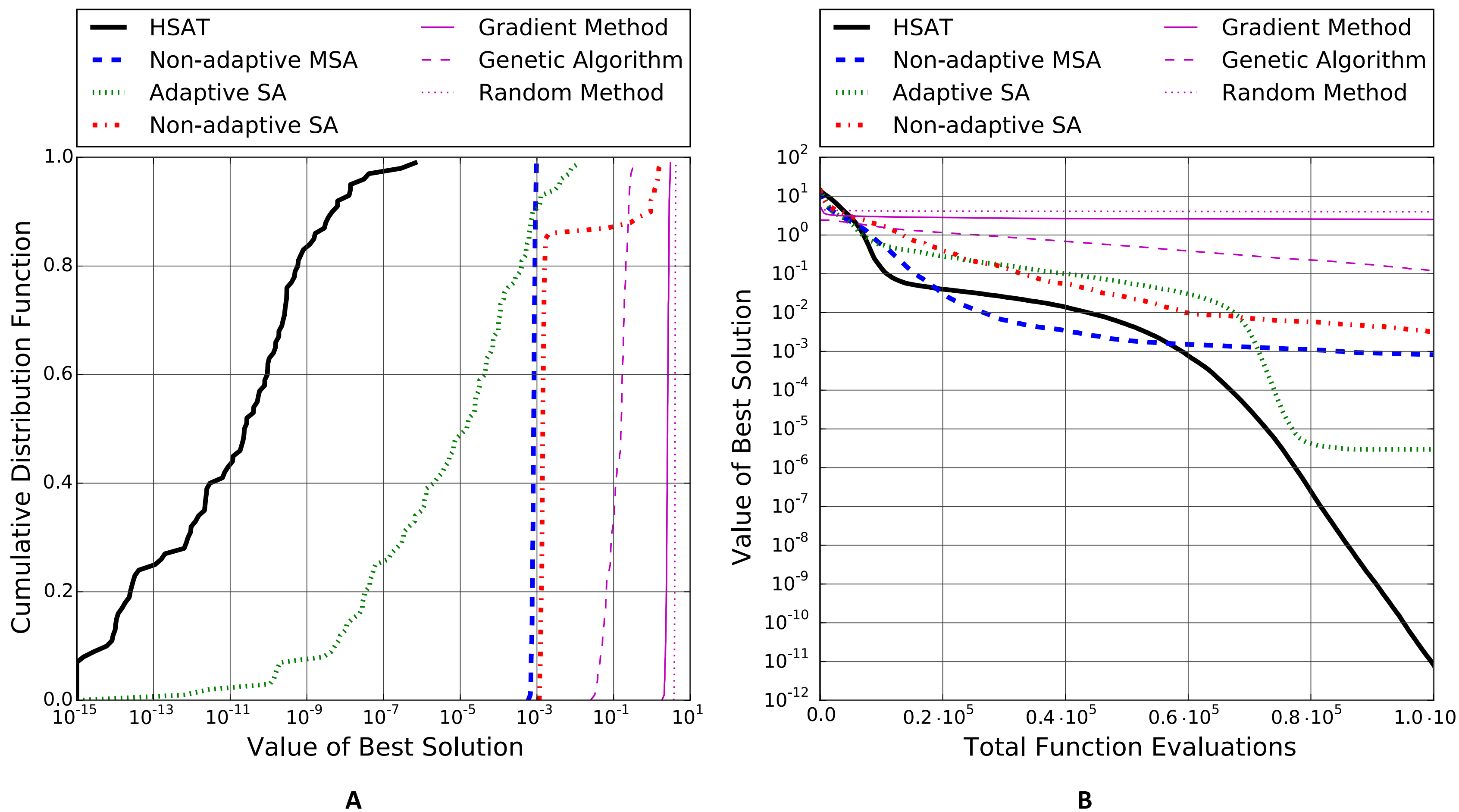
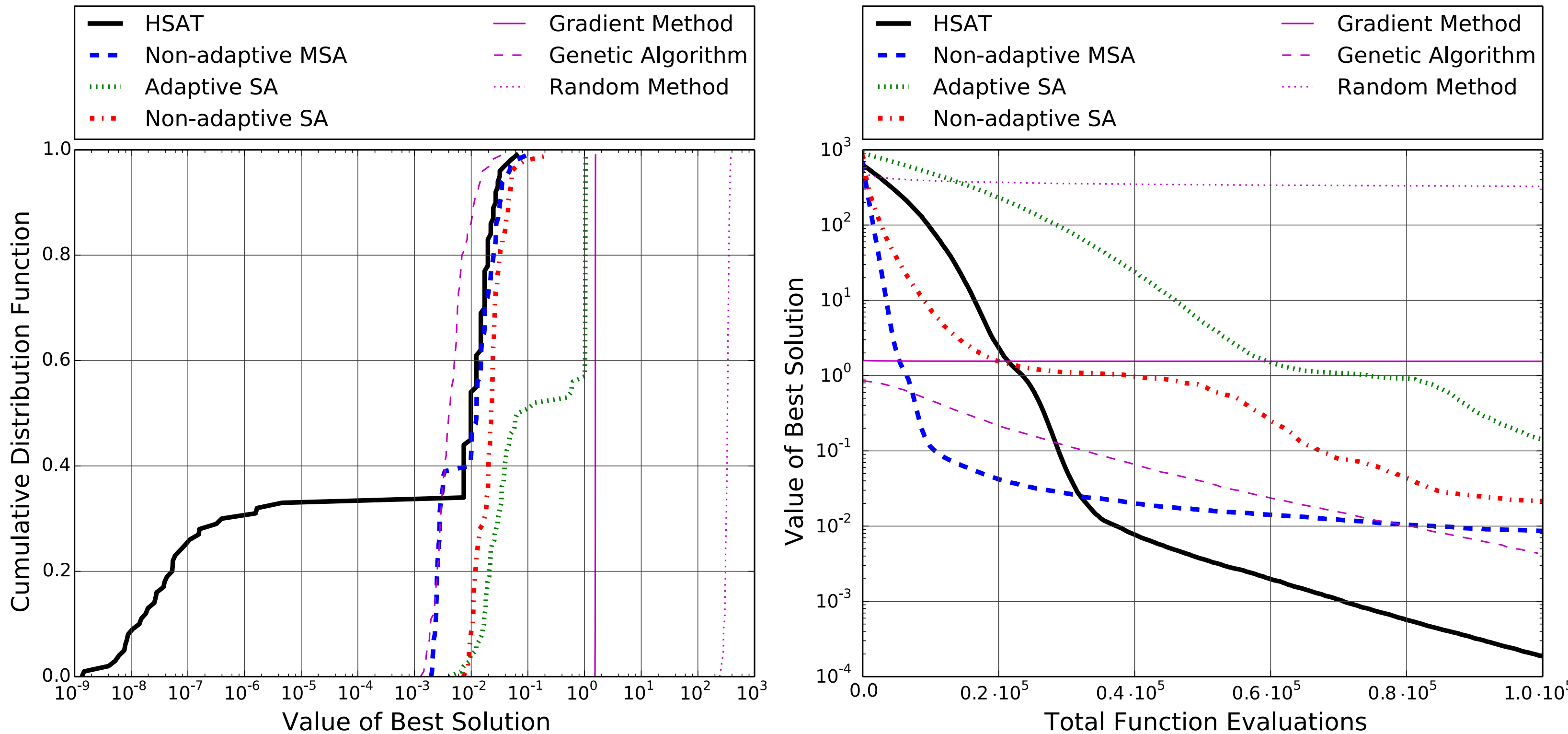


FIGURE 6.6. COMPARISON OF OPTIMIZATION RESULTS FOR ACKLEY FUNCTION SHOWING (A) CUMULATIVE DISTRIBUTION OF TERMINAL SOLUTIONS AND (B) GEOMETRIC MEAN OF BEST SOLUTION FOUND OVER TIME. ERROR BARS OMITTED FOR CLARITY. FOR A GIVEN VALUE ON THE X-AXIS, THE CUMULATIVE DISTRIBUTION FUNCTION GIVES THE FRACTION OF SOLUTIONS THAT ARE AS GOOD AS OR BETTER THAN THAT VALUE.



A **B**

FIGURE 6.7. COMPARISON OF OPTIMIZATION RESULTS FOR GRIEWANK FUNCTION SHOWING (A) CUMULATIVE DISTRIBUTION OF TERMINAL SOLUTIONS AND (B) GEOMETRIC MEAN OF BEST SOLUTION FOUND OVER TIME. ERROR BARS OMITTED FOR CLARITY. FOR A GIVEN VALUE ON THE X-AXIS, THE CUMULATIVE DISTRIBUTION FUNCTION GIVES THE FRACTION OF SOLUTIONS THAT ARE AS GOOD AS OR BETTER THAN THAT VALUE.

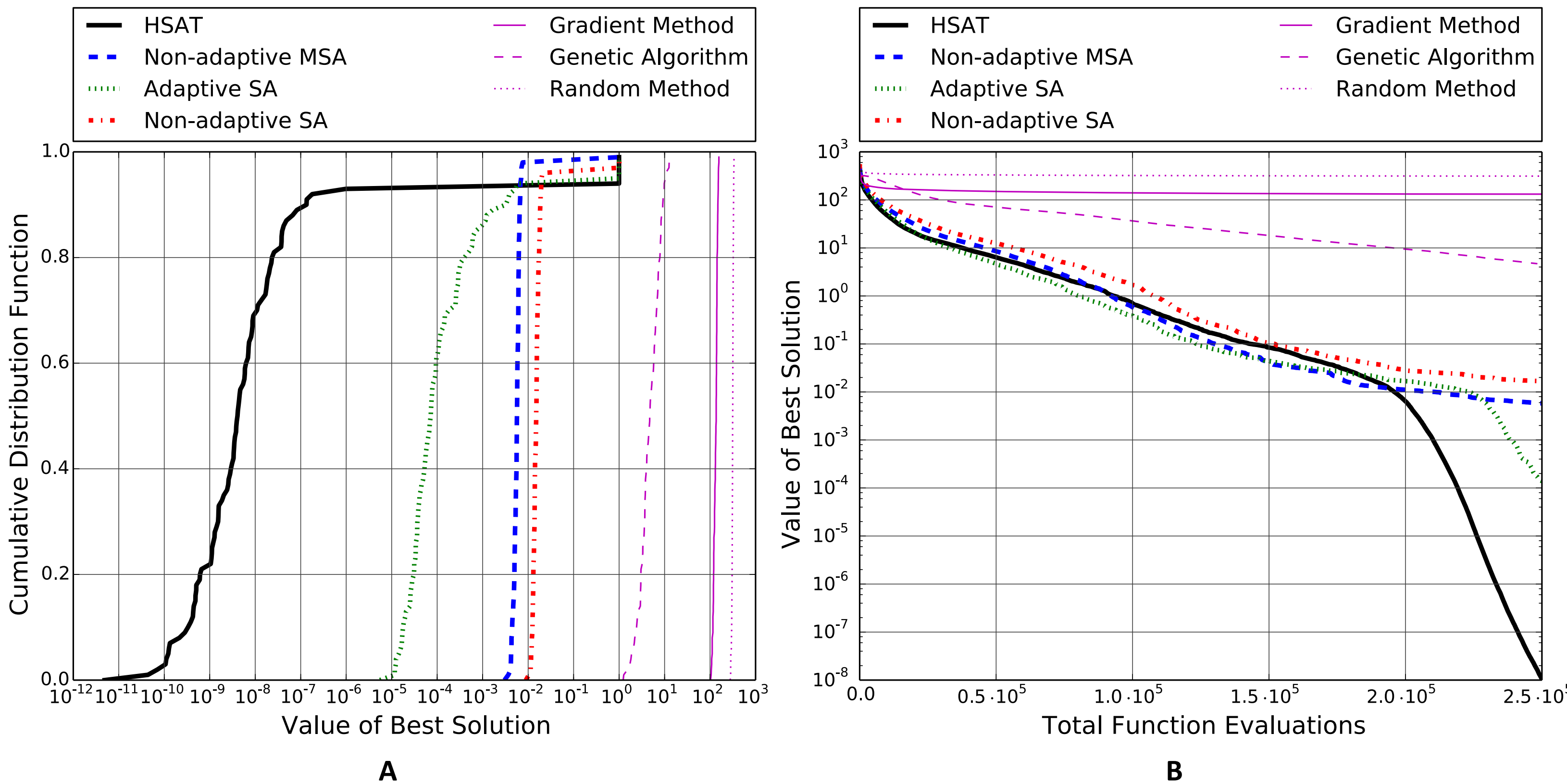


FIGURE 6.8. COMPARISON OF OPTIMIZATION RESULTS FOR RASTRIGIN FUNCTION SHOWING (A) CUMULATIVE DISTRIBUTION OF TERMINAL SOLUTIONS AND (B) GEOMETRIC MEAN OF BEST SOLUTION FOUND OVER TIME. ERROR BARS OMITTED FOR CLARITY. FOR A GIVEN VALUE ON THE X-AXIS, THE CUMULATIVE DISTRIBUTION FUNCTION GIVES THE FRACTION OF SOLUTIONS THAT WERE AS GOOD AS OR BETTER THAN THAT VALUE.

For the Ackley function, all SA-based algorithms outperform the non-SA-based methods (gradient-based algorithm, random search algorithm, and genetic algorithm). Further, there is a significant amount of differentiation amongst the SA-based algorithms evaluated in this chapter. The team-inspired HSAT algorithm returns the best mean final result, besting the others by several orders of magnitude. Examining Figure 6.6.a indicates that *every* HSAT result is better than the best result of most other algorithms, even amongst non-SA-based algorithms (with Adaptive SA being the sole exception). Further, although Non-Adaptive SA and Non-Adaptive MSA return similar result, Non-Adaptive MSA is capable of avoiding some of the local minima that Non-Adaptive SA falls prey to. This can be attributed to the breadth search abilities provided by multiple agents. From the Ackley results in Figure 6.6 it also becomes apparent that the algorithms that utilize adaptive annealing schedules (HSAT and Adaptive SA) are able to obtain the lowest objective function values for the Ackley function. This characteristic is important because algorithms must be capable of transitioning from broad search to downhill optimization upon locating the well in which the global minimum is located.

Results pertaining to the Griewank function are shown in Figures 6.7.a and 6.7.b. In terms of mean performance, the HSAT algorithm provides a better average terminal solution than all other algorithms. However, an examination of Figure 6.7.a shows that in more than 60% of runs the genetic algorithm produces results that are slightly better than HSAT. However, HSAT produces results that are orders of magnitude better than the genetic algorithm in nearly 40% of runs. In contrast to the results on the Ackley function, the highest-performing SA-based algorithms on the Griewank function are those that employ multiple interacting agents. The benefit of multiple interacting agents is attributed to the topography of the Griewank function in the vicinity of the global minimum. Although the global behavior of the Griewank function is approximately convex, the topography near the global minimum is highly multi-modal, and displays a large number of local minima that are near one another in objective function value. The ability to pursue multiple solutions simultaneously is a crucial construct for effectively searching these minima. In the HSAT algorithm, this ability is provided through the use of multiple, independent agents.

The results for the Rastrigin function are provided in Figures 6.8.a and 6.8.b. Similar to the results for the Ackley function, all SA-based algorithms outperform all non-SA-based algorithms. Figure 6.8.a shows that the Non-Adaptive SA and Non-Adaptive MSA algorithms both have similar cumulative distribution functions, and the cumulative distribution of Adaptive SA is slightly better. However, the solutions produced by the HSAT algorithm are shifted significantly to the left, indicating that it returns solutions of higher quality. This indicates that neither the use of interacting agents nor independent adaptive annealing schedules alone leads to better solutions to the Rastrigin function; only the combination of both features improves performance. The unique benefit of this combination can be explained by the topography of the Rastrigin function. Unlike the Ackley and Griewank functions, in which the global behavior is not significantly obscured by the local minima, the local minima of the Rastrigin function almost completely obscure any global behavior. Therefore, multi-agency is beneficial to explore many of these local minima, while individual adaptivity is necessary to ensure a quick transition from broad search of the space to deep search of a specific minimum.

A common trend across all benchmarking functions is that the HSAT algorithm rarely provides the best early solutions. When solving the Ackley function, HSAT initially outperforms other algorithms for the first 20% of runtime, but is later bested by the Non-Adaptive MSA algorithm for nearly 40% of the runtime. It is only at the end that HSAT once again begins to return the best solutions. Similar behavior is observed for the Griewank function: the Non-Adaptive MSA algorithm initially provides the best results before being overtaken by HSAT. For the Rastrigin function all SA-based algorithms provide very similar performance for the majority of runtime. It is only in the last 20% of runtime that HSAT once again exceeds the other SA-based algorithms. The slow initial progress of HSAT is likely due to a broad search phase that does not result in immediate improvement of the solution. Although this phase is of little immediate benefit, it allows agents to locate fruitful regions of the design space so that a deep search is likely to find the global minimum. For problems that are highly multi-modal (such as the continuous numerical functions used in this chapter), larger numbers of function evaluations are required to find the global minimum, which suggests the potential for superior overall performance by HSAT. However, the time required for the broad search phase is not always available during design, so other algorithms (such as a Non-Adaptive MSA) may be preferable if time is limited.

The results of the SA-based algorithms on the benchmarking functions allow us to gain insight into the effect of the two human team characteristics implemented in HSAT (independently controlled adaptive annealing schedules, and probabilistic agent interaction). The Ackley function has a large number of local minima that are similar in objective function value, while the global minimum is contained within a central well. The algorithms that employ an adaptive annealing schedule perform best because they are able to transition from a broad search for the central well to a quick descent towards the bottom of the well. In contrast, the Griewank function has convex global behavior, but in the vicinity of the global minimum there are a large number of local minima with similar objective function values. Therefore, the use of multiple interacting agents becomes crucial to success. The Rastrigin function combines the challenges of both the Ackley and Griewank functions, requiring any algorithm to search of a number of local minima before beginning a deep dive to the global minimum. For this reason, only the combination of interacting agents and adaptive annealing schedules leads to high performance.

These results demonstrate that the unique combination of features found in the HSAT algorithm boosts performance. Much like design teams that are capable of solving a variety of problem types, the features of HSAT ensure high performance across a variety of multi-modal function topographies.

6.6 Summary

In this chapter the CISAT modeling framework, which was originally developed to simulate the behavior of engineering teams, was customized to produce HSAT, an SA-based numerical optimization algorithm. This serves as a demonstration of the versatility of the CISAT modeling framework. Comparing the results of the various algorithms implemented in this chapter provides insight into the performance of HSAT, and the specific benefit derived from both of the team-based characteristics.

The Ackley function has a large number of local minima that are similar in objective function value, while the global minimum is contained within a central well. The algorithms that employ an adaptive annealing schedule perform best because they are able to transition from a broad search for the central well to a quick descent towards the bottom of the well. In contrast, the Griewank function has convex global behavior, but in the vicinity of the global minimum there are a large number of local minima with similar objective function values. Therefore, the use of multiple interacting agents becomes crucial to success. The Rastrigin function combines the challenges of both the Ackley and Griewank functions, requiring any algorithm to search of a number of local minima before beginning a deep dive to the global minimum. For this reason, only the combination of interacting agents and adaptive annealing schedules leads to high performance. These results demonstrate that the unique combination of features found in the HSAT algorithm boosts performance. Further work should use a wider array of benchmarking functions, and vary the dimensionality of those functions. It may also be promising to extend HSAT by structuring the interaction between agents to reflect lessons learned from multi-team organizations.

As implemented, this algorithm could be utilized by engineers and designers to optimize highly-multimodal parametric design problems; one such application could be layout problems which are known to have fractal-like qualities [132]. With minor changes to how solutions are operated on, HSAT could be modified to solve discrete optimization problems as well.

Chapter 7: Investigating Sequences of Action and Strategy in Engineering Design[5]

Nature uses only the longest threads to weave her patterns, so that each small piece of her fabric reveals the organization of the entire tapestry.

Richard Feynman

7.1 Overview

Design often involves searching for a solution by iteratively modifying and adjusting a current design. Through this process designers search the design space, attempting to iteratively improve on their current solution. However, through the process of searching, designers also learn how to efficiently navigate the design space through the operations that they use to modify their solutions. This work studies the ability of designers to learn and apply beneficial sequences of operations, and examines the performance implications of such behavior.

This chapter specifically builds upon an observation made in Chapter 3. In that chapter, it was hypothesized that the order in which operations were performed in that study may have had a large impact on the quality of solutions. By focusing on the ordering of operations, the analysis in this chapter is conducted at a more fine-grained resolution and at shorter timescales than other work on design stage or task sequencing. This degree of resolution is more akin in scope to

---

[5] A portion of this chapter is based on:

McComb, C., Cagan, J., and Kotovsky, K., 2016, "Utilizing Markov chains to Understand Operation Seequencing in Design Tasks," Design Computing and Cognition Conference, Chicago, IL, USA.

studies conducted in the psychology literature. It is at this level that engineers and designers directly interact with potential solutions, so the selection and application of the best actions is particularly crucial for the creation of high-performing solutions. Three questions are specifically addressed in this work:

1. Do designers employ recognizable sequences of operations when solving a design problem, and if so, how might these sequences be characterized?
2. Is the employment of sequences of operations beneficial to designers?
3. Is the order in which operations occur indicative of higher-level strategies?

This work proposes that Markov chains and hidden Markov models can be used as tools for understanding and simulating the order in which humans perform operations in a design context, thus helping to address the three research questions. It should be noted that Markov models do not directly extract fixed-length sequences of operations. Instead, such models implicitly and probabilistically represent the sequences in which operations are likely to occur.

After a brief background section, the main body of this chapter is presented as three investigations, each of which is aligned with one of the research questions posed previously. The first explores whether or not designers employ recognizable operational sequences, and how those sequences might be characterized if they exist. To answer that question, a statistical analysis is conducted on the human data from the cognitive studies of Chapters 2 and 5. This analysis indicates with high confidence that participants in both studies utilized sequences of operations when creating solutions. The results also show that operation sequencing in both studies is best characterized as a first-order Markov process. Higher-order Markov models provide more complex representations but display accuracy that is statistically equivalent to first-order models. The second investigation analyzes whether or not sequences of operations are beneficial to the designer. This is accomplished by performing computational simulations of design team behavior within the Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework, an agent-based platform that approximates the process and performance of engineering design teams (presented in Chapter 3). The insight that human operation sequencing can be treated as a first-order Markov process is used to provide CISAT agents with the ability to learn sequences, enabling a computational comparison between teams with and without sequence-learning abilities. These simulations demonstrate that the ability to learn sequences during design was extremely beneficial to teams in the cognitive studies. The third and final investigation searches for higher-level strategic patterns in the sequencing of design operations using hidden Markov models. It is shown that participants in both studies proceeded through similar patterns of topological construction, spatial negotiation, and detailed optimization.

7.2 Background

The ability to learn sequences of operations is essential to human performance across a variety of both everyday and specialized tasks [195]. A related behavior is the ability of humans to collect many pieces of related information within a single unit of memory. This process is commonly referred to as *chunking*, and has been observed across a variety of domains [196–198]. Observations of chunking behavior in humans have even led to the enhancement of computational design algorithms [199]. While chunking is an important aspect of design, this work focuses on an equally important aspect, the sequencing of operations. Sequencing may be thought of as temporal chunking, but the two behaviors are functionally distinct.

7.2.1 Sequence Learning

In some domains, the presence of sequential behavior has been shown to be indicative of expertise [95]. However, in other studies of individual problem-solving that used the Tower of Hanoi puzzle [56,57] or the Thurstone Letters Series Completion task [200] it was shown that participants were able to acquire and begin employing move sequences within a short period of time. In studies using the Tower of Hanoi puzzle, participants spent most of their time learning to sequence moves appropriately, and a relatively short amount of time in finally solving the puzzle [56]. When comparing easy and difficult isomorphs of that puzzle, it was discovered that participants solving hard isomorphs spent longer in the learning phase, but that both conditions took approximately the same amount of time in the final solving phase [56]. Studies using the Thurstone task identified that participants' problem-solving behavior consisted broadly of two steps. In the first step participants recognized some order in the letter series and codified it as a rule (i.e., generating a pattern), and in the second step they used that rule to extrapolate the letter series (i.e., generating a sequence) [200]. This two-step process was validated with computer simulations [200], and has since been revisited and confirmed [201]. Other work explicitly studied the order of operations used in a geometric analogy task [202]. Although the task itself placed no constraints on the order in which operations could be performed, it was found that there was a strong preference amongst participants for a specific order [202]. Further, participants made to use a non-preferred order performed with lower speed and accuracy, indicating that the preferred order was an important construct [202].

There is evidence that humans learn sequences implicitly, without the need for direct attention [25,26] but there is also evidence that explicitly learning sequences can boost effectiveness [23,24]. This evidential duality gives credence to the theory that sequence learning can occur through a variety of cognitive pathways [195,203].

7.2.2 Sequencing in Design

Research on sequencing in design has examined the ordering of design stages, the ordering of specific tasks, and the sequencing of discrete design operations. These three types of design sequences can be conceptualized along a continuum that describes the degree of abstraction, from design stages (the most abstract and generalized of the three) to design operations (the least abstract and most specific to the task at hand). These types of sequences can also be differentiated in terms of the timescale at which the sequenced object is enacted, with design stages at the longest timescales and design operations at the shortest.

The sequencing of design stages is commonly examined through design studies with either teams or individuals. In a study that coded intra-team design communication as either content-focused or process-focused, it was noted that teams had a high probability of remaining on one focus for several communicative acts before switching to the other focus [75]. The same study also coded communication in terms of steps in the design process, and noted a similar pattern of repeated communicative acts within a step [75]. Another study tasked participants with the design of a playground, and activity was coded with respect to design stages [204]. Expert sequences flowed smoothly from one stage to the next, while novice sequences were more choppy and unstable [204]. In a comparison of undergraduate and PhD students, it was noted that there was significant variability in the sequence in which design stages were visited [205]. Interestingly, none of the participants followed a linear design process [205]. Yet a similar study showed that participants who transitioned linearly through the design process produced more effective solutions [206].

The sequencing of individual design tasks however takes place at shorter timescales than the sequencing of design stages, with numerous design tasks usually performed within a single stage. Intelligent ordering of tasks can increase the concurrency with which tasks can be completed [207], minimize the time and cost involved in developing a product [208], and increase the availability of information for important decisions [209]. Waldron and Waldron observed the sequence of tasks involved in the design of a complex walking vehicle [210]. They noted that there was not a distinct separation between conceptual and detailed design and that tasks may span across design stages [210]. Theoretical work on task sequencing has shown that optimal strategies for ordering tasks may be tied to problem complexity [209]. Other work has utilized genetic algorithms to optimize task sequences for a variety of different objectives [211].

The sequencing of discrete design operations takes place on the shortest timescales and has the most intimate impact on potential solutions. It is at this level that engineers and designers directly interact with solution attributes, so selection and application of the best operations is particularly crucial for the creation of high-quality solutions. Much of the work that has examined the sequencing of design operations makes use of some type of protocol encoding scheme in order to render the resulting sequences meaningful. Function-Behavior-Structure (FBS) concepts [212,213] are commonly used to create coding schemes to study sequencing at this scale. The FBS design ontology specifically describes design as a process with the ultimate goal of transforming a set of design requirements into a design description [212]. The description cannot proceed directly from the set of requirements, but instead arises as a result of considering issues associated with the design requirements – the required functionality, the expected behavior, the observed behavior, and the structure of the designed object. The transitions between these issues are referred to as processes. First-order sequential behavior in the FBS ontology (the transitions between issues) has been modeled via Markov chains [214–216]. Second-order sequential behavior (the probability that specific processes will precede specific issues) has also been investigated [217]. Simulations have also considered the effects of memory on sequencing behavior in computational agents using higher-order Markov chains [218]. Aside from studies of human designers, the extraction and implementation of beneficial rule pairs has been explored with respect to design automation [219].

This work will also use Markov concepts to study the ordering of operations during design tasks. Instead of using a coding scheme that requires human assessment, we code design operations according to their quantitative effect on the current design solution. Markov chains concepts are used to study the sequencing of operations for design tasks, and also to implement sequence-learning abilities within CISAT, the computational model of design teams introduced in Chapter 3. Hidden Markov models are also used to describe higher-level sequencing of designer behavior, extracting behavioral constructs that align with designer intent. Both Markov chains and hidden Markov models are introduced in the next section.

7.2.3 Markov Processes

A Markov process is a stochastic process in which a system transitions between a finite number of discrete states. Markov processes are commonly modeled using Markov chain models [220]. In a first-order Markov chain model, the probability of transitioning to a future state depends only on the current state of the system, and not on previous states [220]. The probability of transition to a future state from a current state is given by the transition matrix, $\boldsymbol{T}$, where the value of T_{ij} is the probability of transitioning from state i to state j. Markov chains were first introduced more than a century ago [221], and have since been used in a number of applications including computer performance evaluation [222], web search [223], chemical processes [224], and design team communications [225,226].

Figure 7.1 depicts a first-order Markov chain with three states (S_1, S_2, and S_3). The entries of the transition matrix are shown next to arrows that indicate the relevant transition. Self-transitions are allowed in the model, meaning that the system remains in the same state across multiple time steps. In this work, Markov chains are used to describe the order in which modifying operations are applied to designs. Operations constitute the states of the Markov chain model, and the transition probabilities in $\boldsymbol{T}$ describe the probability that one operation will be followed by another.

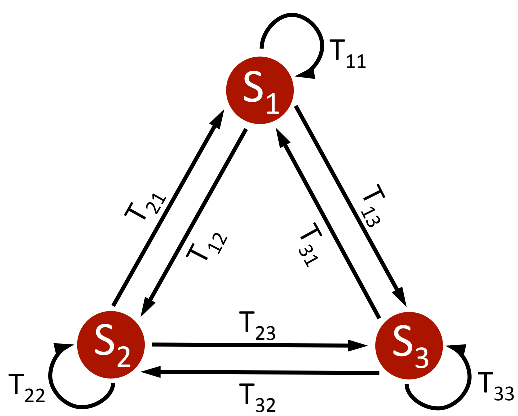


FIGURE 7.1. EXAMPLE MARKOV CHAIN WITH 3 STATES.

Higher-order Markov chain models assume that the selection of the next state is dependent on the current state as well as some number of past states, thus encoding some degree of "memory" in the model [227]. Within the domain of design, the inherent memory of higher-order models could provide a useful analog of the expertise and memory of human designers. Higher-order Markov chain models are used in this work to characterize how much inherent memory is necessary to describe the order in which designers apply modifying operations to a solution concept. Zero-order Markov chains are also used in this work. These models do not encode a sequential representation of data, but instead encode the non-conditional frequency with which operations are applied (much like the probabilities associated with each side of a weighted die).

Using Markov chains of any order to model a Markov process assumes that the states of the model can be observed directly. However, it may be the case that the states themselves are not directly observable. Hidden Markov models are based on this assumption, and instead assume that the hidden states can only be observed by proxy through their probabilistic emission of tokens. The transitions that occur between states are given by a transition matrix, $\boldsymbol{T}$, and the probabilities governing the emission of tokens from the hidden states are given in an emission matrix, $\boldsymbol{E}$. The mathematics describing hidden Markov Models were established by Baum and colleagues [228–231]. They have since been applied to fields as diverse as speech recognition [232], protein modeling [233], economic forecasting [234], and team military tactics [235].

Figure 7.2 shows a hidden Markov model with 3 hidden states and 3 emission tokens. Hidden states are shown by circles, and emission tokens are shown with squares. The transitions between hidden states are shown with solid arrows, and the emissions are shown with dashed arrows. The entries of the emission matrix are shown next to arrows that indicate the relevant emission, and transition probabilities are omitted for clarity.

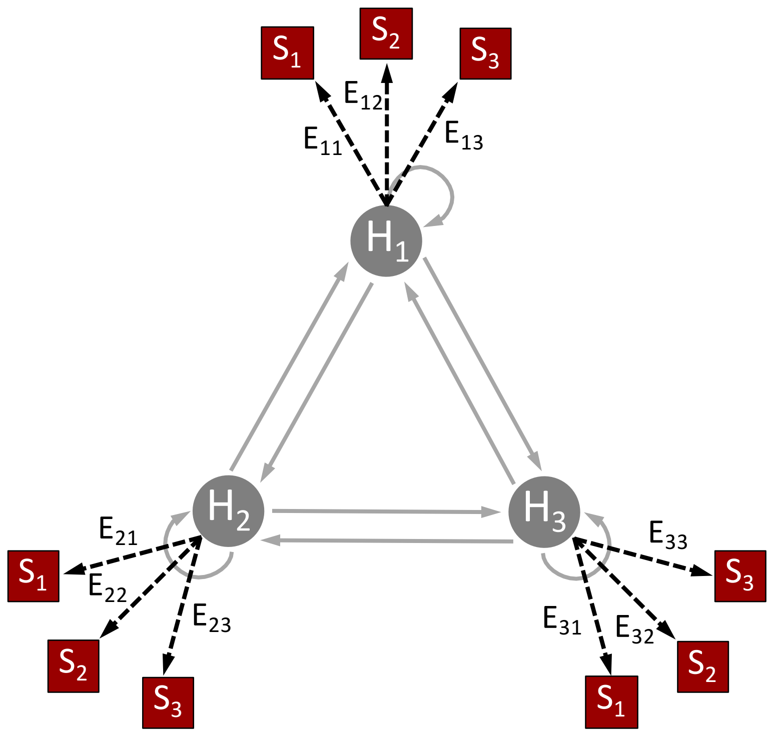


FIGURE 7.2. EXAMPLE HIDDEN MARKOV MODEL WITH 3 HIDDEN STATES AND 3 EMISSION TOKENS.

In the application explored in this work, the tokens emitted by the model are design operations. By treating the design operations as probabilistic representations, the hidden states that constitute the model become the underlying cognitive or strategic states that the designer goes through.

7.3 Data Sets

The operation data sets analyzed in this chapter were derived from the cognitive studies introduced earlier in the dissertation. The first, which will be referred to as Study A, tasked engineering students with designing a truss and was introduced in Chapter 2. The second study, which will be referred to Study B, tasked a different group of engineering students with the design of an internet-connected home cooling system and was introduced in Chapter 5. A brief review of both studies is given here.

7.3.1 Data from Study A: Trusses

Study A tasked 16 small teams of 3 engineering design students with the design of a truss structure. Design was conducted over the course of six, 4-minute design sessions. During the study, the design problem was changed twice without warning through the introduction of new problem statements. The initial problem statement simply asked teams to design a truss with two loading points and three supports. The first change presented participants with the same layout, but also instructed them to account for the fact that any one of the supports could be removed at any time. The second change modified the problem so that teams had to build their truss around an obstacle. Teams were given a target mass and factor-of-safety for the initial problem statements and for each of the subsequent modified problem statements. Because the problem statement changed during design, it is expected that this data set will yield sequencing information that applies generally to a variety of truss design problems.

To facilitate design, every participant was given access to a computer-implemented truss design program that was created for the purpose of the study. This program allowed the participants to build, assess, and share truss designs within their teams. In addition to facilitating design, the truss design program was used to continuously record the operations that participants chose and applied in order to modify their designs, thus allowing a full account of design operations to be accumulated. The operations available to participants were the following:

1. Adding a joint
2. Removing a joint
3. Adding a member

4. Removing a member
5. Changing the size of a single member
6. Changing the size of all members
7. Moving a joint

This information was post-processed to extract a string of move operators (denoted by the integers 1 through 7) for each of the 48 participants in the study. A typical solution sequence consists of 400 to 500 operations. A short example operation sequence is provided in Figure 7.3.

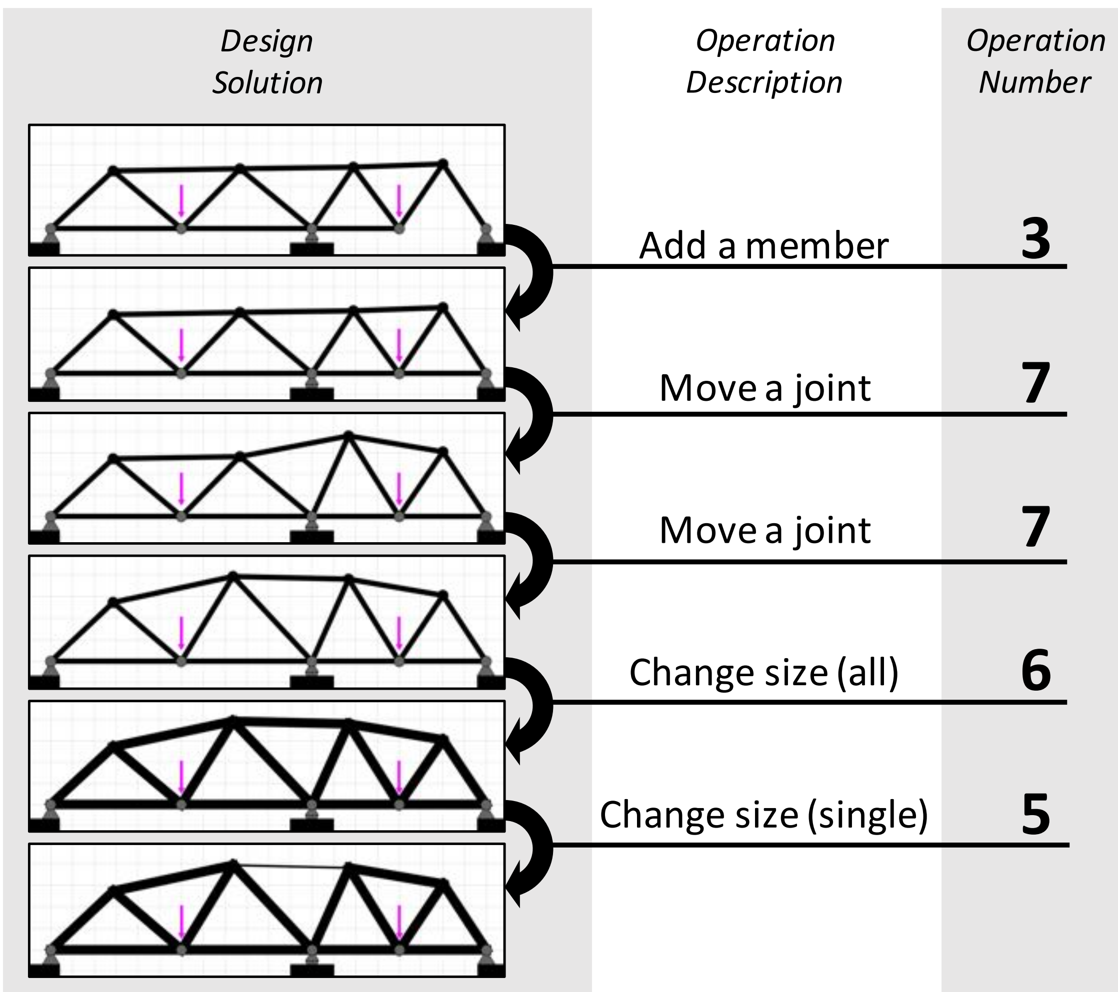


FIGURE 7.3. EXAMPLE TRUSS OPERATION SEQUENCE, WITH OPERATION NUMBERS CORRESPONDING TO THE LIST OF TRUSS OPERATIONS.

7.3.2 Data from Study B: Home Cooling Systems

Study B tasked 54 mechanical engineering students (either individually or in teams) with designing a system of connected products to maintain the temperature within a residential structure. Participants were allowed to use and connect three distinct product types to create their solutions: sensors (which sensed the temperature of the room in which they were placed), coolers (which cooled rooms in the home), and processors (which made decisions about which coolers to activate based on information from sensors). Design was conducted over the course of a 30-minute session, during which participants were allowed to perform only 50 design operations.

Participants were also given access to a computer-implemented design program in this study. This program allowed the participants to build, assess, and share solutions, and it was also used to continuously record the operations that participants used. The operations available to participants were the following:

1. Add processor
2. Add sensor
3. Add cooler
4. Remove processor
5. Remove sensor
6. Remove cooler
7. Move sensor
8. Move cooler
9. Tune cooler

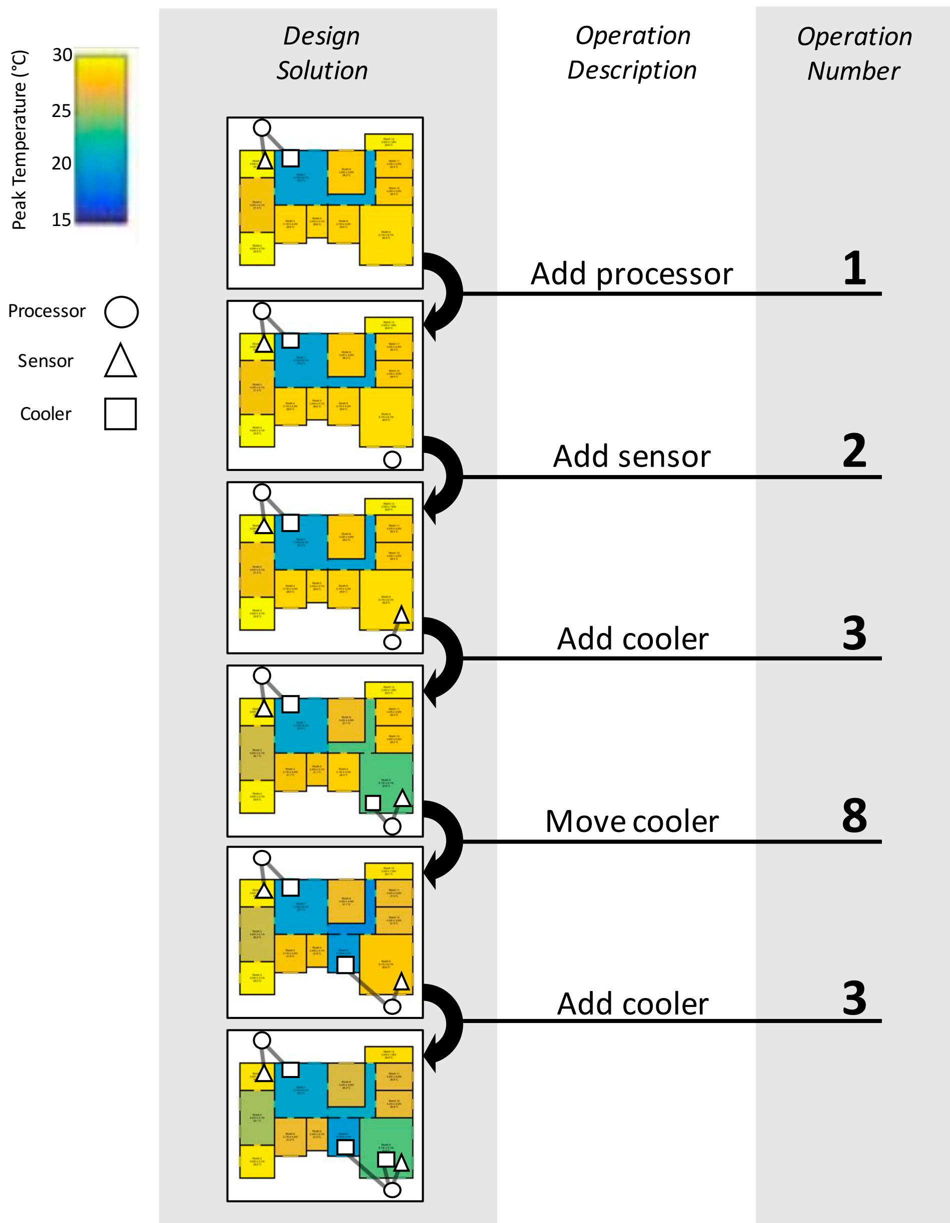


FIGURE 7.4. EXAMPLE COOLING SYSTEM OPERATION SEQUENCE, WITH OPERATION NUMBERS CORRESPONDING TO THE LIST OF SYSTEM OPERATIONS. COLORS INDICATE ROOM TEMPERATURES.

This information was post-processed to extract a string of move operators (denoted by the integers 1 through 9) for each of the 54 participants in the study. Every solution sequence consisted of exactly 50 operations. A short example sequence is provided in Figure 7.4. The solution diagrams in the left column depict a plan view of the structure, with colors indicating the relative temperature in different rooms.

7.4 Use of Operation Sequences

This chapter first analyzes human operation data from the two design studies with the objective of determining whether or not designers employed operation sequences, and if so, what degree of order is necessary to accurately represent the data.

7.4.1 Methodology

Markov chain models were trained on the human data in order to provide a statistical representation of the sequence in which operations occurred in the cognitive study. The following discussion of the training process is based on material in [220], but is presented in terms of design operations (rather than Markov process states). The procedure specifically outlines the training of first-order Markov models, but it can also be applied to higher-order models with small modifications.

A Markov chain is defined by the values of its transition matrix, $\boldsymbol{T}$. Element T_{ij} in the matrix defines the probability that the next operation will be operation j, given that the previous operation was operation i. Therefore, the elements of the transition matrix for a Markov chain can be estimated based on observed data using

$$T_{ij} = \frac{N_{ij}}{N_i} \qquad 7.1$$

where N_{ij} is the number of times that operation j is observed to follow operation i, and N_i is the number of times that operation i is observed.

The log-likelihood is a quantity that indicates the probability that a model could have produced a given set of data, and can thus be used to compare different models. This is essentially a measure of the accuracy with which the model can reproduce a given data set. The log-likelihood for a Markov chain model ($\mathcal{L}_{MC}$) is

$$\mathcal{L}_{MC} = \sum_{i=1}^{M} \sum_{j=1}^{M} N_{ij} \cdot ln\,(T_{ij}) \qquad 7.2$$

where N_{ij} and T_{ij} are defined as above and M is the number of different operations.

The procedure for training higher-order Markov chains follows essentially the same pattern as that for first-order Markov chains. The key difference is that the states of the model are no longer single design operations, but n-tuples of operations, where n is the order of the Markov chain. Training of a zero-order Markov chain model simply consists of estimating the frequency with which each operation occurs, without any assumption of conditional dependence on earlier operations in the sequence.

Markov models were trained on both data sets, from zero order (i.e. a model assuming that future operations have no dependence on past operations) to fourth order (i.e. a model assuming that future operations dependent on the last four operations). Models were trained using leave-one-out cross-validation [236]. For a data set consisting of n samples, this cross-validation approach trains a model with $n - 1$ samples, and then tests the model on the sample that was not used for training. This procedure is repeated until every individual sample has been used for testing.

7.4.2 Results for Study A

A plot of log-likelihood for models of increasing order is shown in Figure 7.5, with error bars indicating standard error. The dashed line shows the log-likelihood of the model on the training data set, and the solid line shows the log-likelihood on the testing data set. Significant differences between adjacent models are shown with dotted brackets. Models with higher testing log likelihood provide a better fit to unseen data, and should be preferred.

Figure 7.5 shows a steep increase in log-likelihood from the zero-order Markov model (which by nature cannot model any sequential behavior) to the first-order Markov model (which provides the simplest representation of sequencing behavior). This indicates that strong sequencing patterns do exist in the data from the truss study. However, after first-order the testing log-likelihood plateaus, while the training log-likelihood continues to increase. This indicates that overfitting occurs in higher-order models. In other words, higher-order models begin to fit attributes of the training data that are not general, and thus exhibit lower accuracy on the testing data set.

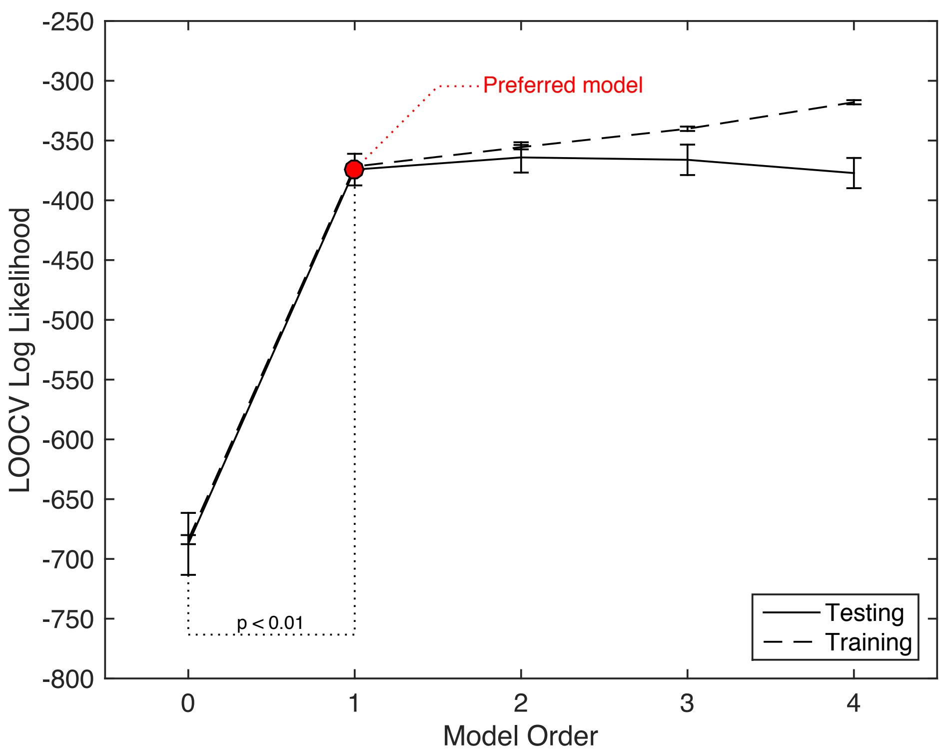


FIGURE 7.5. LOG-LIKELIHOOD COMPUTED ON STUDY A DATA FOR MODELS OF INCREASING ORDER. ERROR BARS SHOW ±1 S.E. AND P-VALUES INDICATE THE RESULTS OF ANOVA TESTS BETWEEN THE TESTING LOG-LIKELIHOOD OF CONSECUTIVE MODELS.

As noted previously, the testing log-likelihood in Figure 7.5 plateaus after first-order, with no further significant differences apparent in the testing log-likelihood of the models. Therefore, the first-order model is preferred, as it provides a degree of accuracy that is statistically equivalent to higher order models, but it does so with a much lower degree of complexity. Designer activity in the truss design task can therefore be accurately modeled as a first-order Markov process. Any increase in accuracy provided by higher-order models is statistically insignificant.

Comparing the zero-order model (which encodes a non-sequential representation) to the first-order model (which encodes a representation of designer activity that is both simple and accurate) enables an examination of why sequencing of operations was important for the truss design task. The operation frequencies associated with the zero-order Markov model are shown in Figure 7.6, and the transition matrix of the first-order Markov chain model is shown in Figure 7.7. The colors of the squares indicate the magnitude of the probability, which is also noted numerically within each square.

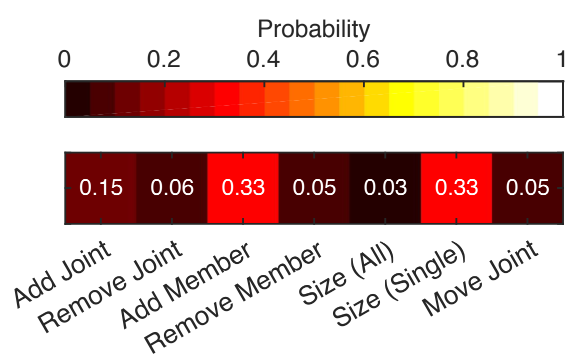


FIGURE 7.6. STATE FREQUENCIES FOR ZERO-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY A.

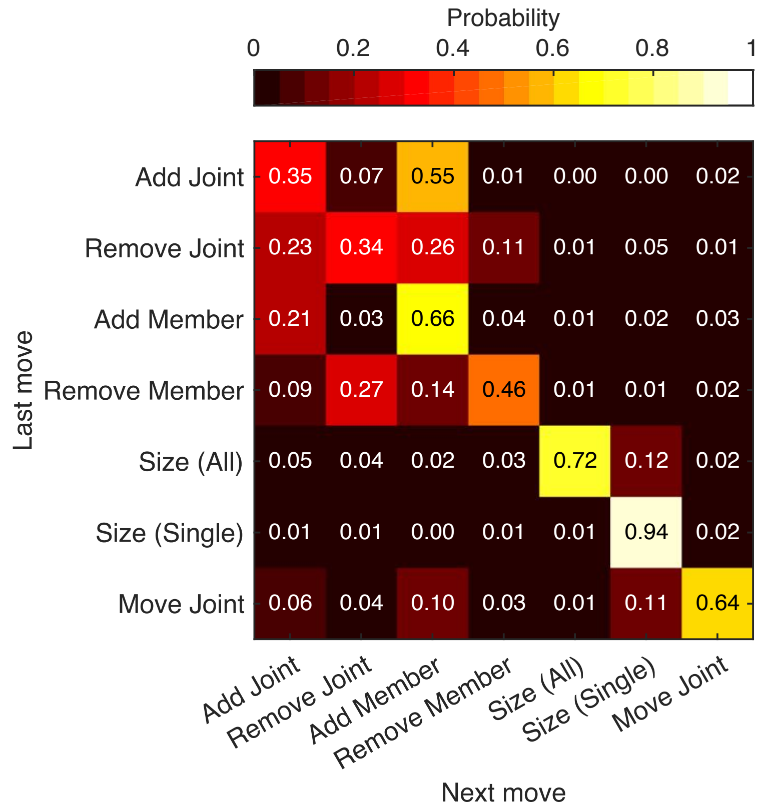


FIGURE 7.7. TRANSITION MATRIX FOR FIRST-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY A.

Examining the differences between Figures 7.6 and 7.7 provides some insight as to why the first-order model is so much more veridical than the zero-order model. The most striking aspect of the transition probability matrix of the first-order model is that it is largely diagonal. This indicates that the same operation was likely to be applied several times before a different operation was selected. This aspect of the participants' behavior simply could not be captured in the zero-order model. For instance, the zero-order model indicates a 33% chance that the next operation will be to add a member, regardless of the previous operation. However, the column of the first-order transition matrix that corresponds to adding a member shows that that operation is most likely chosen after adding a joint, removing a joint, or adding a member, and rarely selected if the last move involves changing the size of truss members. Thus, the selection of the subsequent operations is highly conditional on the last operation performed. Similarly, the zero-order model shows a 33% chance of changing the size of a single member. However, the first-order transition matrix more incisively encodes the operational data by showing that that operation is rarely selected after adding a joint, removing a joint, adding a member, or removing a member.

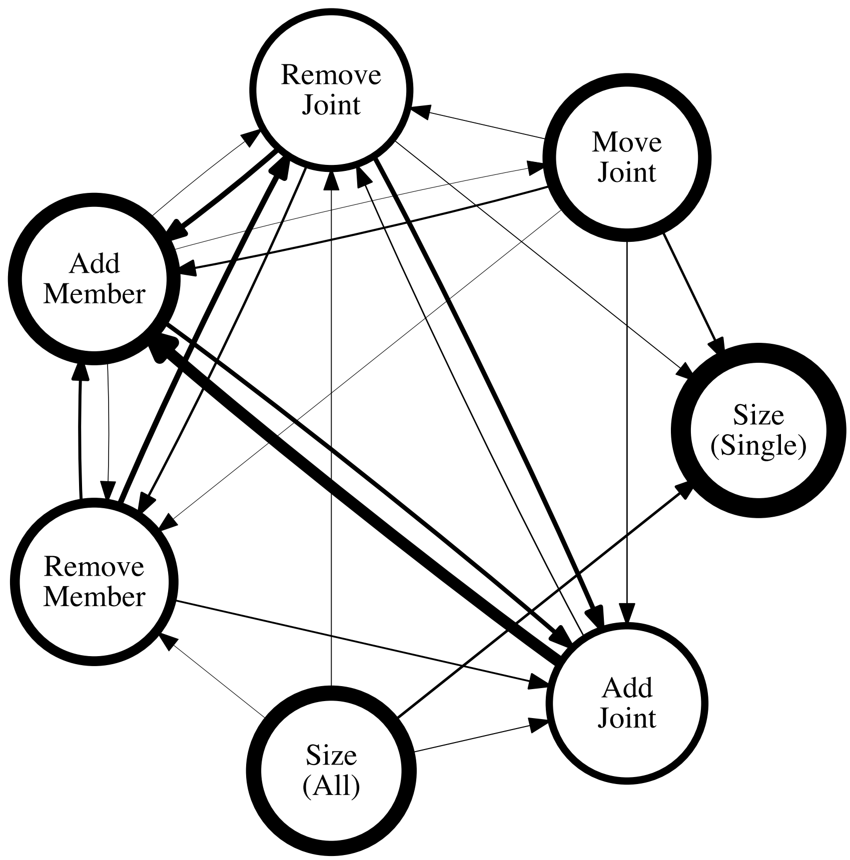


FIGURE 7.8. GRAPHICAL REPRESENTATION OF THE FIRST-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY A SHOWING THE MOST LIKELY TRANSITIONS. TRANSITION PROBABILITY IS INDICATED BY THE LINE THICKNESS OF THE CORRESPONDING ARROW. IN THE CASE OF SELF-TRANSITIONS, THE PROBABILITY IS INDICATED BY THE THICKNESS OF THE BORDER OF THE CORRESPONDING OPERATION.

In addition, a graphical representation of the first-order Markov chain model is provided in Figure 7.8. Arrows are used to indicate transitions between states and line thicknesses represent the relative likelihood of those transitions, with thicker lines indicating the more likely transitions. Only the strongest transitions were included in the representation (those with transition probabilities above the median, approximately 0.03). The probabilities pertaining to self-transitions are indicated by the thickness of the border of the corresponding operation. Figure 7.8 provides further insight as to the pattern of operations in the study. Actions involving the topology of the truss (adding and removing joints and members) are connected by the thickest arrows, indicating that there is a strong probability of transition between these operations. This indicates that these topology operations were usually employed together during the design study. In contrast, operations that only change the shape of the truss (changing the size of members, or moving joints) are connected by arrows that are much thinner, indicating that there is far less interaction between these operations. These operations are also much more likely to be applied multiple times in a row, as indicated by the probability of a self-transition. Finally, there is a very low probability of leaving the state corresponding to changing the size of a single member ("size (single)"). This indicates that participants were very likely to repeatedly apply this operation to fine-tune the size of the members in their designs.

The first-order Markov model assumes that the selection of a subsequent operation is dependent on only the last operation, so that each operation is causally linked to the next. Sequences of arbitrary length can be created by stringing together several of these causal links. Graphically, the process of constructing these sequences consists of tracing a path through the graph-based representation of the transition matrix shown in Figure 7.8. A set of high likelihood exemplar sequences produced through this process is provided in Figure 7.9. Operations are shown in rectangular boxes and the probability that the following operation would occur is given with a percentage over the linking arrow.

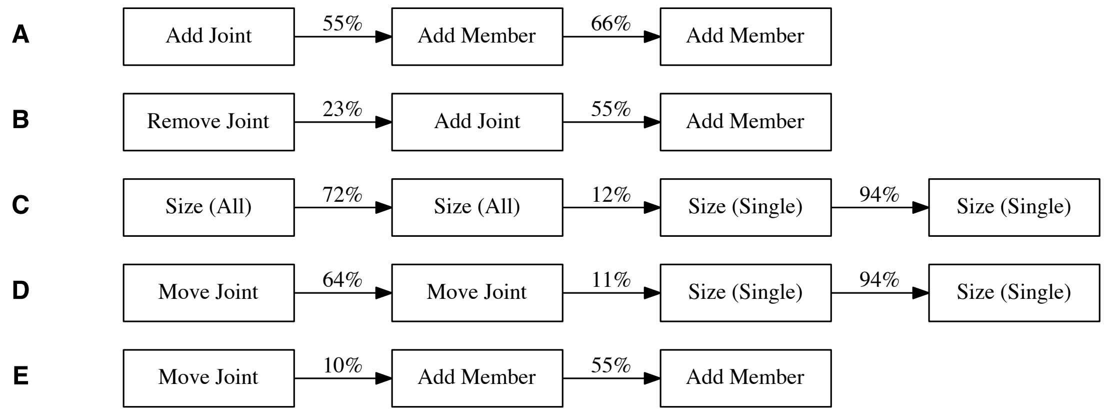


FIGURE 7.9. EXEMPLAR SEQUENCES EXTRACTED FROM FIRST-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY A.

These sequences are multi-operation patterns of action that might be expected in a truss design task. Sequence A consists of a joint addition followed by several member additions. This kind of pattern could arise as a designer constructs their truss, adding a joint and then attaching it to the existing truss with new structural members. Sequence B is similar in that it consists of topology operations, but instead begins with a joint removal (which also removes all attached members) following by a joint addition and a member addition. This signifies revision of a section of the truss – removing a section of the truss with poor performance and then rebuilding it in an attempt to improve performance characteristics. Sequences C and D define procedures for fine-tuning a fixed truss topology – joint repositioning or the adjustment of global members sizes, followed by the adjustment of the size of specific members. Sequence E describes a return from shape optimization to topology optimization – the repositioning of a joint (possibly to make room for new truss elements) followed by the addition of new structural members.

7.4.3 Results for Study B

The same methodology used for Study A was applied to operation data from Study B. A plot of the log-likelihood for models of increasing order is provided in Figure 7.9. The dashed line shows the log-likelihood of the model on the training data set, and the solid line shows the log-likelihood on the testing data set.

Many of the same trends from Figure 7.5 are echoed here. There is an increase in log-likelihood between the zero-order and first-order Markov models, indicating that sequencing behavior is evident in the study data. After first-order, the training log-likelihood continues to increase, while the testing log-likelihood first plateaus (first- to second-order) and then decreases (second-order and higher).. Once again, this indicates that higher-order models tend to overfit the training data, losing generalizability.

The first-order Markov model is preferred for this design task; it offers a degree of accuracy that is better or statistically equivalent to higher-order models, but does so with much lower model complexity. Recall that the first-order Markov model was also preferred for study A. This may indicate that the sequencing of design operations can generally be treated as a first-order Markov process for the types of configuration tasks examined in the chapter. At the very least, it is evidence that lower-order processes tend to be the most veridical. Higher-order models may learn specific sequential constructs that are informative, but they do not appear to offer a superior description of general patterns of design activity.

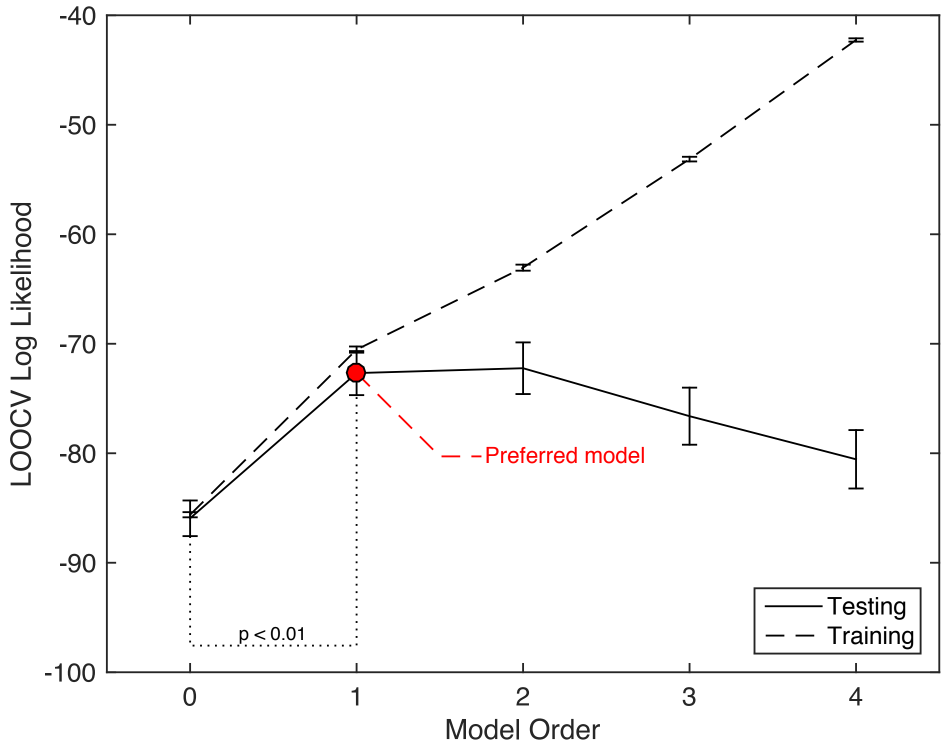


FIGURE 7.10. LOG-LIKELIHOOD COMPUTED ON STUDY B DATA FOR MODELS OF INCREASING ORDER. ERROR BARS SHOW ±1 S.E. AND P-VALUES INDICATE THE RESULTS OF ANOVA TESTS BETWEEN THE TESTING LOG-LIKELIHOOD OF CONSECUTIVE MODELS.

Examining the differences between the zero-order Markov model (non-sequential) and first-order Markov model (which provides a sequential description similar to that used by designers in the study) can once again provide insight as to the benefit that is derived from pursuing operations sequentially for the design task in Study B. The zero-order Markov probabilities are shown in Figure 7.11, and the transition matrix for the first-order Markov model is shown in Figure 7.12. The colors of the squares in these figures indicate the magnitude of the probability, which is also noted numerically within each square.

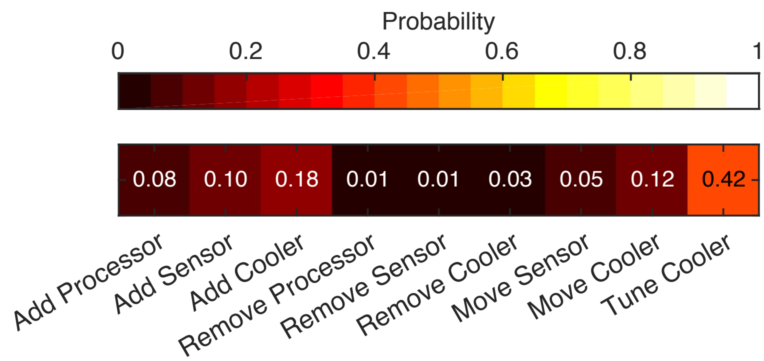


FIGURE 7.11. STATE FREQUENCIES FOR ZERO-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY B.

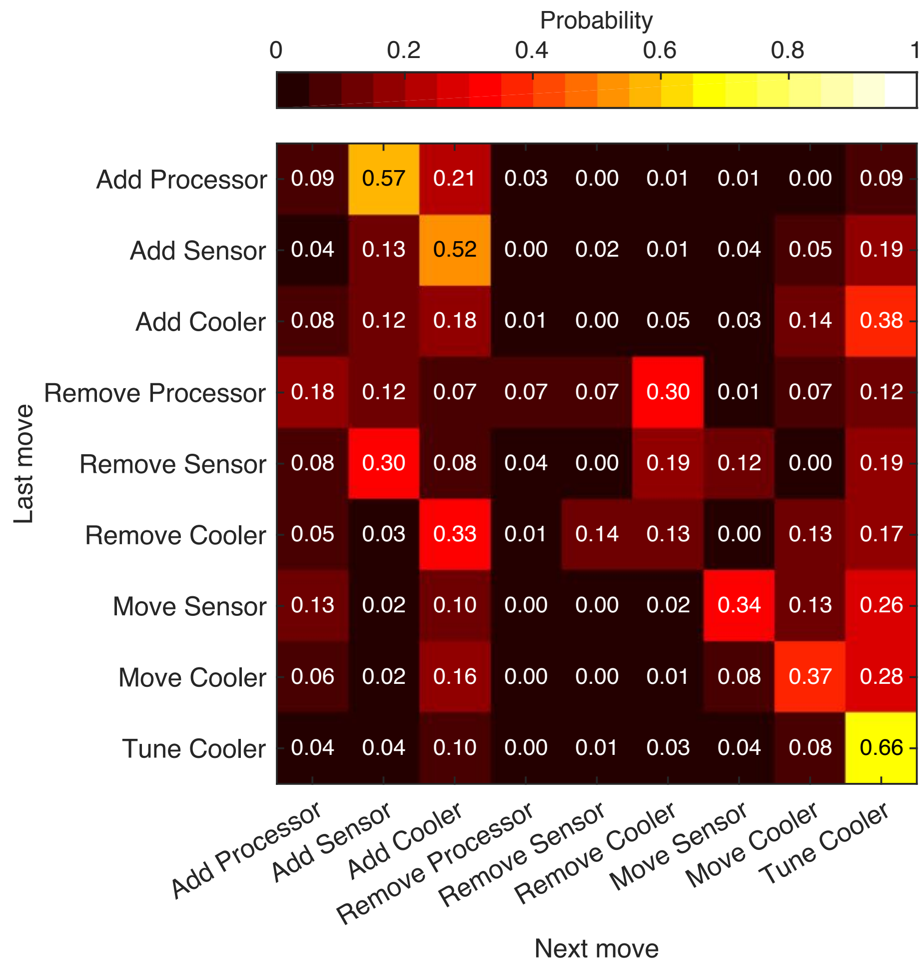


FIGURE 7.12. TRANSITION MATRIX FOR FIRST-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY B.

Whereas the first-order Markov model developed for Study A had a strongly diagonal structure, the transition matrix from Study B has a number of strong off-diagonal elements and relatively weak diagonal elements. This indicates that there is little propensity to apply the same operator multiple times in a row. The only operations with more than a 30% chance of being applied multiple times in series are sensor movement, cooler movement, and cooler tuning. It should be noted that these are the shape operations. In contrast, the topology operations (adding or removing products) have lower probabilities of being applied multiple times in series.

A graphical version of the first-order Markov transition matrix is provided in Figure 7.13. The graph is thresholded in a way that is similar to Figure 7.8 so that only the most likely transitions are shown. This representation reinforces many of the trends observed in the raw transition matrix. Two trends are made particularly clear in this graph. The first is the highly probable linkage from adding a processor, to adding a sensor, to adding a cooler. This sequence enables the construction of the simplest independent subsystem possible, consisting of a sensor to read the temperature in a room, a cooler to act on the temperature in a room, and a processor to decide when to activate the cooler based on information from the sensor. The second trend is the strong connectedness of the cooler tuning operation to nearly every other operation. This indicates that cooler tuning played an integral role in the production of solutions, and was frequently utilized throughout the design process.

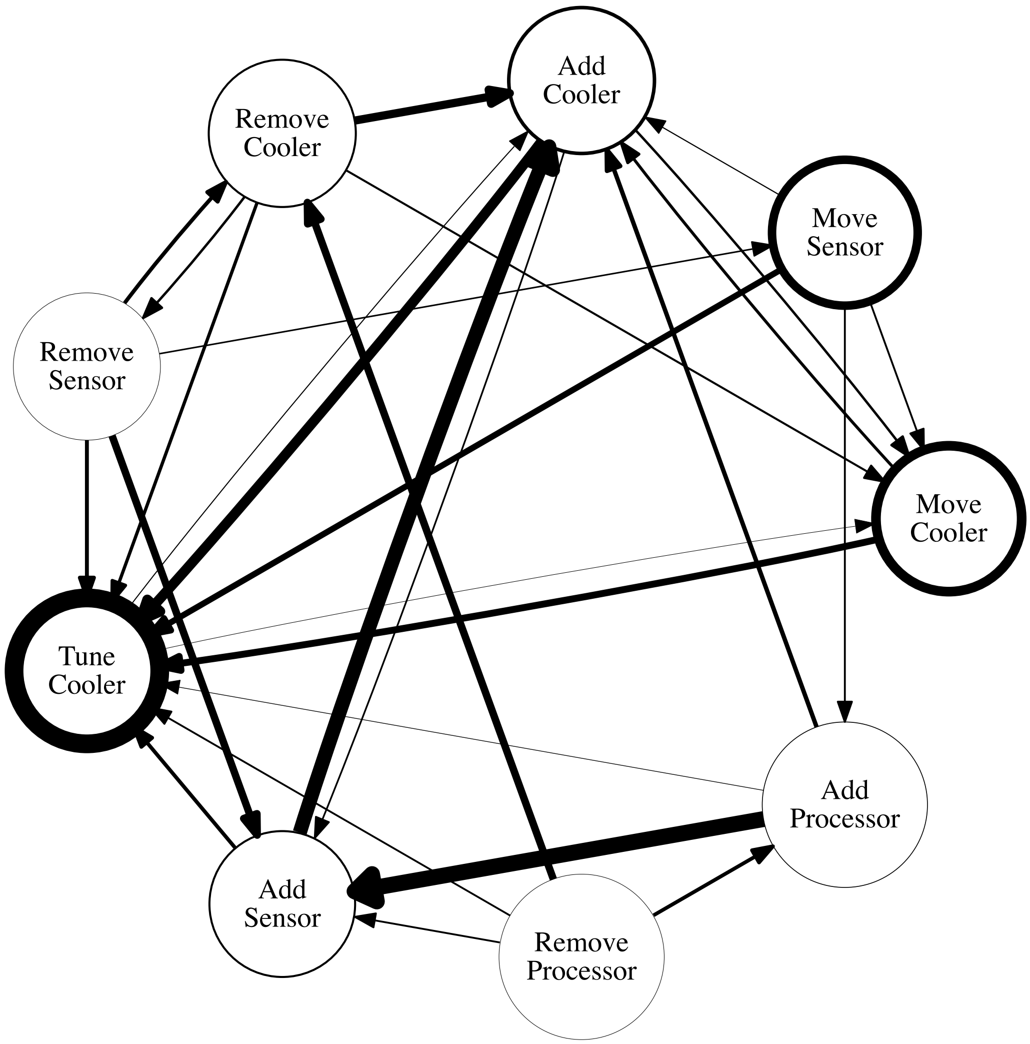


FIGURE 7.13. GRAPHICAL REPRESENTATION OF THE FIRST-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY B SHOWING THE MOST LIKELY TRANSITIONS. TRANSITION PROBABILITY IS INDICATED BY THE LINE THICKNESS OF THE CORRESPONDING ARROW. IN THE CASE OF SELF-TRANSITIONS, THE PROBABILITY IS INDICATED BY THE THICKNESS OF THE BORDER OF THE CORRESPONDING OPERATION.

The first-order Markov model assumes that temporally adjacent operations are causally linked. Longer sequences of operations can be extracted from the first-order transition matrix by stringing together several of these causally linked operations. The procedure of extracting longer sequences is graphically equivalent to traversing the graph-based representation of the transition matrix (see Figure 7.13). In this case, some of the potentially meaningful sequences are 3 or 4 operations long. A set of some of the most likely operation sequences is provided in Figure 7.14.

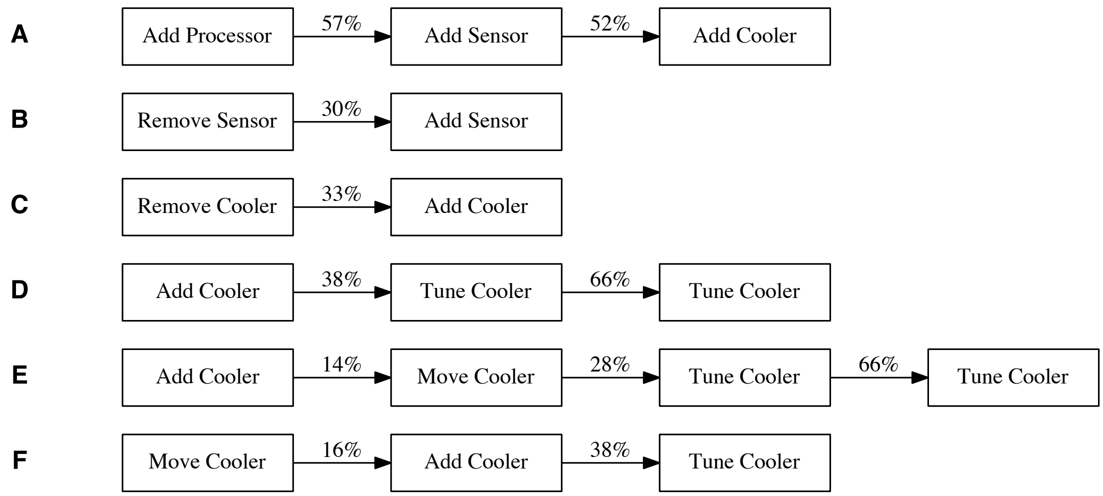


FIGURE 7.14. EXEMPLAR SEQUENCES EXTRACTED FROM FIRST-ORDER MARKOV MODEL LEARNED ON DATA FROM STUDY B.

Sequence A consists of a processor addition, a sensor addition, and a cooler addition. As noted above in Figure 7.13, this sequence encodes the construction of the simplest independent subsystem possible, consisting of a sensor, a cooler and a processor. Sequences B and C describe the removal followed by the addition of either a sensor or a cooler, respectively. These actions were necessary to transfer the control of a cooler or sensor to a different processor – this was not enabled with a single move during the study. Interestingly, the probability of the opposite of these two sequences (adding a sensor/cooler and then deleting a sensor/cooler) was nearly 0. Sequences D, E, and F are all sequences related to placing and modifying coolers. The prevalence of these sequences might be expected since the operations for adding and tuning coolers were applied the most often (see Figure 7.11). Sequence D describes the common action sequence of adding a cooler and then immediately tuning its properties. An alternative cooler-related sequence is presented with Sequence E in which a cooler is added, moved to a new location, and then tuned. This sequence would have been enacted when the cooler did not function as expected where it was placed. Sequence F is related to sequence E in that it consists of the same operations but they are enacted in a different order. Sequence F begins with moving a cooler, an action that might leave part of the building under-cooled. A new cooler is then added (ostensibly in the under-cooled area) and tuned to optimize performance.

7.4.4 Discussion

This section analyzed data from two design studies by fitting Markov models of increasing order to the operational data from the study. These models progressively encoded greater degrees of memory, meaning that the choice of the next operation was based on a greater number of prior operations. Two important findings resulted from this analysis.

1. It is likely that participants in both studies utilized operation sequences.
2. Designers' operational sequences can be modeled accurately using first-order Markov chains; higher-order Markov chains do not lead to significant increases in accuracy.

The first finding stems from a comparison of the zero- and first-order Markov models. Zero-order Markov models cannot encode sequence information, while first-order Markov models provide a minimal representation of sequencing, in which selection of the next operation is conditional upon only the last operation. For both studies, first-order Markov models fit the operation data better than zero-order models, thus demonstrating that operation sequencing is evident.

The second finding stems from a comparison of the first-order and higher-order Markov models. The first-order Markov model provided a fit that was either equivalent to or better than the higher-order models for both studies. Higher-order models encode sequences that are dependent on multiple prior operations, instead of just the most recent single operation.

Therefore, the higher fit of the first-order model indicates that memory of multiple past operations is not necessary to accurately model the selection of future operations.

These first-order Markov models assume that a designers' choice of a subsequent operation is dependent only on what the last operation was, establishing a causal link between the two. By stringing together several of these causally linked operations, sequences of arbitrary length can be created. The process of creating these long sequences essentially amounts to traversing the graph described by the first-order transition matrix. As demonstrated in Figures 7.9 and 7.14, the longer sequences extracted by this method describe meaningful patterns of design.

7.5 Benefit of Operation Sequences

The previous section established that there are recognizable operation sequences in the design data, and demonstrated that these sequences are best approximated using first-order Markov models. This section investigates whether or not the use of operation sequences was beneficial to the participants in the studies. In other words, we have demonstrated that designers employ operational sequences, but do these sequences actually lead to a better outcome? Sequence learning has been recognized in other domains as a sign of expertise, and may help designers to quickly find fruitful regions of the design space. However, it may also bias designers towards learning sequences that greedily improve solution quality. Applying these greedy sequences could critically limiting breadth of search and lead designers towards local minima of inferior solution quality.

Because sequence learning can take place implicitly [25,26] it would be nearly impossible to control as an experimental variable in a study with human participants. It is possible that implicit learning processes could take over even if participants were successfully prevented from explicitly learning any operation sequences. For that reason, this work utilizes the CISAT modeling framework (introduced in Chapter 3) to test the effects of sequence learning. Individuals and teams simulated in CISAT have clearly defined abilities, making it possible to perfectly control the way that these simulated individuals learn operational sequences. Therefore, a comparison between sequential and non-sequential learning patterns becomes feasible, and offers insight as to the benefit of sequence learning for real human designers.

7.5.1 Methodology

The operational learning characteristics of the CISAT framework is of particular interest in this work. The intention behind this characteristic is to provide CISAT agents with the ability to learn which move operators should be applied in the future. The characteristic was originally implemented as follows. During every iteration, an agent selects which move operator to apply next by taking a random draw from a multinomial distribution defined by a vector of probabilities, $\boldsymbol{p}$. The chosen move operator, i, is then used to modify the current solution. If operator i improves the quality of the solution, then the probability that that operator will be chosen in the future is increased using the equation

$$p_i \leftarrow p_i \cdot (1 + k_{OL}), \tag{7.3}$$

where k_{OL} is a constant used to control how quickly learning occurs. If the move operator worsens the quality of the solution, the probability of selecting it in the future is decreased using

$$p_i \leftarrow p_i \cdot (1 - k_{OL}). \tag{7.4}$$

The probability vector is re-normalized following every update. This is conceptually similar to the operator reinforcement approach used in rule-based interactive tree search [87,88], but it is generally simpler in implementation.

A second version of the operational learning characteristic was implemented using first-order Markov chain constructs. This model was chosen specifically because it was shown in the previous section to accurately encode human operation sequences. This concept is implemented so that agents select move operators to apply using the probabilities in a first-order Markov chain transition matrix, $\boldsymbol{T}$, and then iteratively tune the probabilities in $\boldsymbol{T}$ to bias selection towards the best sequences of move operators. This process is similar to the bipartite pattern/sequence generation process identified by Kotovsky et al. for human participants solving the Thurstone Letter Series Completion task [200]. Here, the tuning of probabilities in $\boldsymbol{T}$ constitutes the pattern generation step, and the selection of a subsequent operation based on $\boldsymbol{T}$ is analogous to the sequence generation step.

To select which move operator to apply, a random draw is taken from the multinomial distribution defined by row i of the matrix $\boldsymbol{T}$, where operator i is the last move operator applied. The chosen operator, operator j, is then applied to the current solution. The values in $\boldsymbol{T}$ are updated depending on whether the quality of the current solution improves

$$T_{ij} \leftarrow T_{ij} \cdot (1 + k_{OL}), \qquad 7.5$$

or worsens

$$T_{ij} \leftarrow T_{ij} \cdot (1 - k_{OL}). \qquad 7.6$$

Through the Markov chain transition matrix, the selection of the next move operator is made conditional upon the last move operator that was applied. This modification enables

CISAT agents to learn and apply beneficial sequences of operations. Note that the first-order Markov chain model does not directly extract fixed-length sequences of operations from the experience of the agent. Rather, these sequences are implicitly coded within the transition matrix of the Markov chain model by updating probabilities according to move operator performance (i.e., whether applying a move operator improved or worsened solution quality). This makes the enactment of previously beneficial sequences more likely, thus accounting for sequence learning within the CISAT modeling framework. However, the probabilistic nature of operator selection still allows for the exploration and discovery of new sequences.

7.5.2 Results for Study A

In the interest of simplicity, performance was only simulated for the initial problem statement from Study A. A total of 100 teams were simulated for each of the two conditions: sequential, utilizing first-order Markov chain concepts; and non-sequential, utilizing zero-order Markov chains. The results of the simulations were then post-processed to track each team's best solution over time. The mean Normalized Strength-to-Weight Ratio (SWR) of each team's current best solution was then computed. This metric serves as an indication of the quality of a structural solution. A comparison of the two conditions is shown in Figure 7.15.

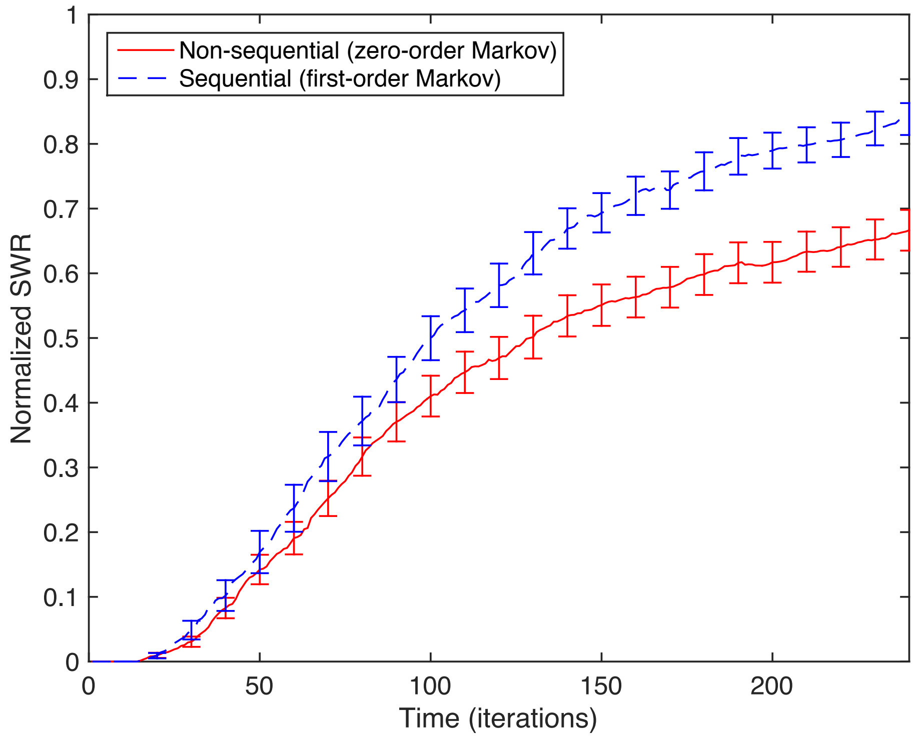


FIGURE 7.15. COMPARISON OF TRUSS DESIGN QUALITY FOR ZERO-ORDER AND FIRST-ORDER MARKOV MODELS, REPRESENTING SEQUENTIAL AND NON-SEQUENTIAL LEARNING APPROACHES. ERROR BARS SHOW ± 1 S.E.

The difference in final design quality between the two conditions is highly significant ($\rho < 10^{-3}$, $F = 11.2$) and the condition using sequence learning achieved a higher final design quality. Since CISAT has been shown to accurately model engineering design teams, these simulations indicate that the ability to learn operation sequences while solving a design problem is beneficial to human designers. The CISAT simulations indicate that operation sequencing was a beneficial aspect of human cognition during the truss design task.

7.5.3 Results for Study B

CISAT was also used to simulate the performance of human design teams on the cooling system design task. Sequential and non-sequential learning was implemented as above with first-order and zero-order Markov chains, respectively. The results of the simulations were then post-processed to track each team's best solution over time. In this case, normalized cooling efficiency was computed for the series of best solutions as an indicator of quality. This metric is the cooling capacity of the system (the extent to which it decreased the peak temperature in the home), divided by the total cost of the system. This ratio was then normalized according to the target values for total cost and peak temperature. A comparison of the mean normalized cooling efficiency of the two conditions is shown in Figure 7.16.

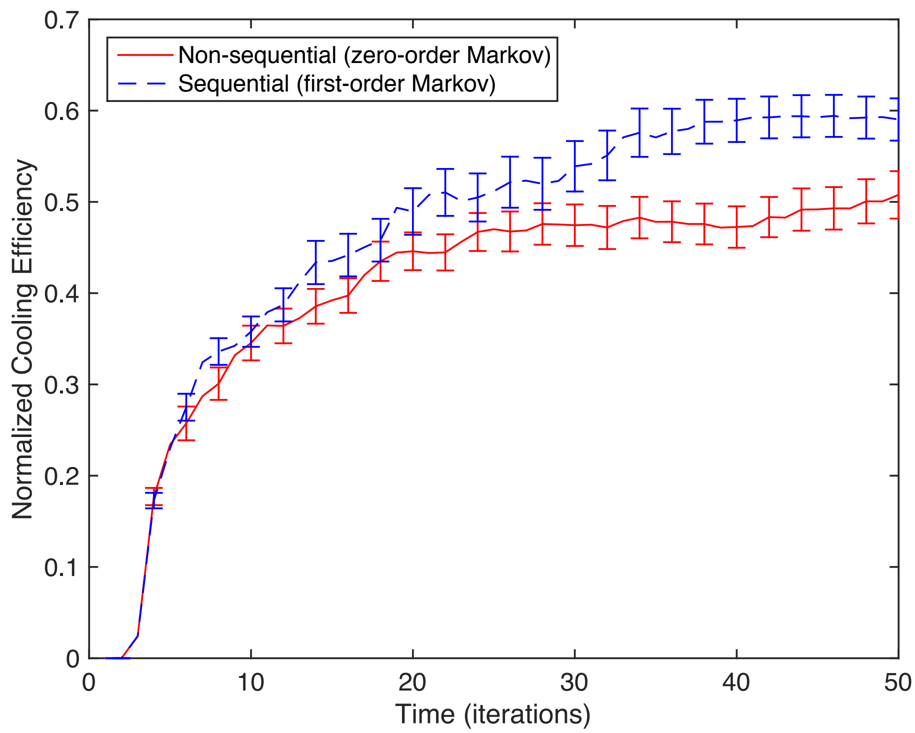


FIGURE 7.16. COMPARISON OF COOLING SYSTEM DESIGN QUALITY FOR ZERO-ORDER AND FIRST-ORDER MARKOV MODELS, REPRESENTING SEQUENTIAL AND NON-SEQUENTIAL LEARNING APPROACHES. ERROR BARS SHOW ± 1 S.E.

For this task, teams in the condition that was capable of learning and employing sequences of operations achieved solutions with significantly higher quality ($\rho < 0.05, F = 5.91$). Since CISAT emulates human design teams, this indicates that sequence learning abilities were beneficial to designers solving the cooling system design.

7.5.4 Discussion

In this section the operational learning characteristic of CISAT was modified to enable agents to learn beneficial sequences of operations by reinforcing a first-order Markov transition matrix. This enabled a comparison between CISAT-simulated teams employing either non-sequential learning (a zero-order Markov model) or sequential learning similar to that observed in designers (a first-order Markov model). Simulations were conducted on the design problem from Study A as well as the problem from Study B. In both cases, sequential learning produced solutions with higher quality. This indicates that sequence learning is a beneficial aspect of human cognition during design. Therefore, designers should emphasize the importance of learning beneficial operation sequences during iterative design problems.

7.6 Higher-level Sequential Strategies

The previous two sections examined the existence and impact of operation sequences, primarily using first-order Markov models. This section entertains the possibility that design operation data may belie higher-level sequences. Rather than treating design operations as the states of a Markov chain, we assume here that design operations are probabilistic representations of a set of hidden states. Through this treatment the hidden states that constitute the model become representative of the underlying cognitive or strategic states that the designer goes through during design.

7.6.1 Methodology

If the number of hidden states (k) is known, a hidden Markov model can be trained using the Baum-Welch algorithm. This algorithm provides a means of estimating the transition matrix (which dictates the transition probabilities between hidden states) and the emission matrix (which defines the distribution of token emissions over hidden states), which describe a hidden Markov model. A mathematical description of the algorithm is not given here, but is it described in detail in [237].

However, the correct number of hidden states to use for the current application in design is not known. Therefore, it becomes necessary to use the Baum-Welch algorithm to train a number of models with varying values of k, and then compare them in some way. In this work we trained several models with values of k from 1 to the number of operations associated with the design problem (7 for Study A, and 9 for Study B). Higher values of k are not necessary because the emission probabilities of the states are no longer linearly independent when the number of hidden states is greater than the number of emission tokens. For each value of k models were trained using leave-one-out cross-validation [236]. For a data set consisting of n samples, this cross-validation approach trains a model with $n-1$ samples, and then tests the model on the sample that was not used for training. This procedure is repeated until every individual sample has been used for testing.

A preferred model was then selected from the set of trained models based on the testing log-likelihood (indicative of its ability to represent data that it was not explicitly trained on). Specifically, we selected the simplest model for which the testing log-likelihood is not significantly different from testing log-likelihood of the most complex model. This selection criterion balances between model parsimony and descriptive accuracy by selecting the model that has the smallest number of hidden states necessary to offer a significantly accurate description of the data.

7.6.2 Results for Study A

Figure 7.17 shows the results of training hidden Markov models with varying numbers of hidden states on the truss design data set (Study A). Models trained with 3 or fewer hidden states had testing log-likelihood values that were significantly lower than the most complex model (7 hidden states in this case), indicating that these models offered relatively low accuracy. The first model that was not statistically different in terms of testing log-likelihood is that trained with 4 hidden states. Therefore, the four-state model is selected as the preferred model in this application.

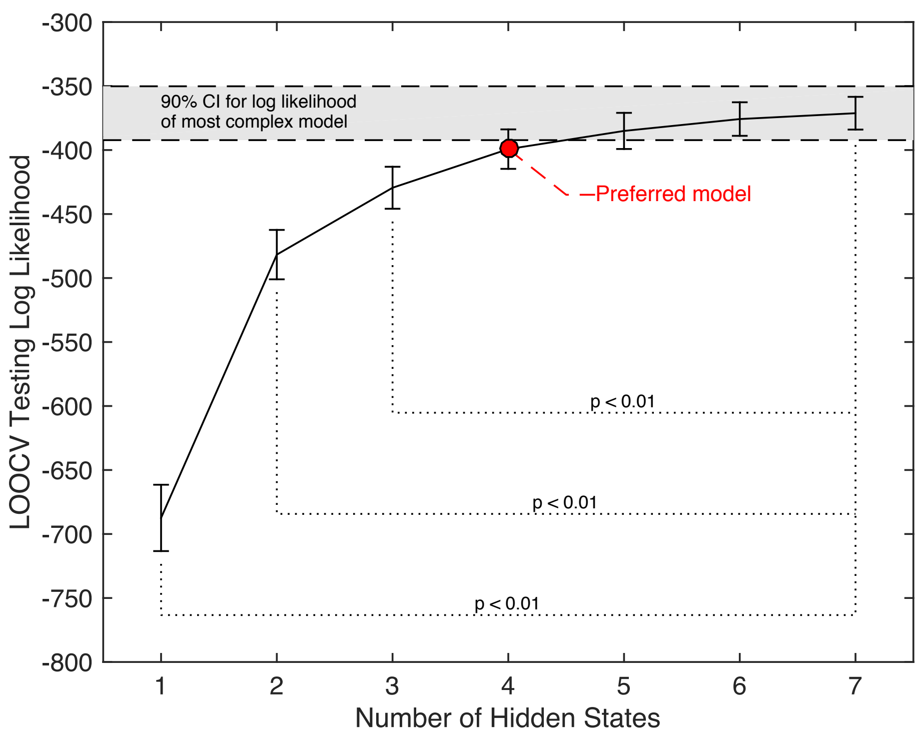


FIGURE 7.17. TESTING LOG-LIKELIHOOD OF MODELS WITH INCREASING NUMBER OF HIDDEN STATES LEARNED ON DATA FROM STUDY A. ERROR BARS SHOW ±1 S.E. AND P-VALUES INDICATE THE RESULTS OF ANOVA TESTS.

The transition and emission matrices associated with this 4-state model are provided in Figure 7.18. The color of each square in the matrices indicates the magnitude of the element, and the probability value is also denoted by the text within the square.

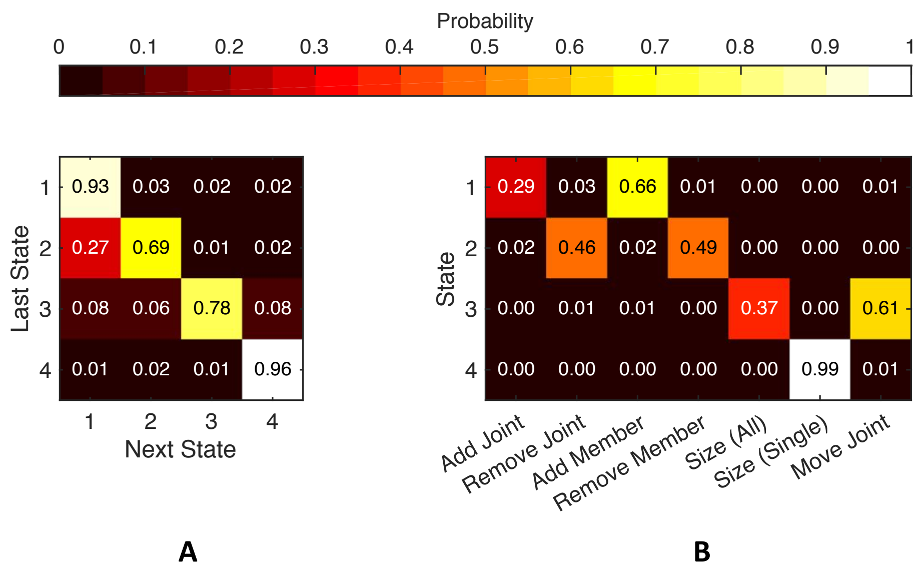


FIGURE 7.18. PARAMETERS OF FOUR-STATE HIDDEN MARKOV MODEL BASED ON DATA FROM STUDY A, SHOWING (A) TRANSITION MATRIX AND (B) EMISSION MATRIX.

The transition matrix of the hidden Markov model (shown on the left in Figure 7.18) has strong diagonal elements, all of which are more than 60% and two of which are above 90%. This indicates that a designer is likely to remain in the same state while applying several successive operations. The operations that a designer applies in a given state are shown in the emission matrix on the right in Figure 7.18. The emission matrix shows that operations are almost perfectly partitioned into states. In other words, a given operation is likely to be applied in only one of the four states.

A visual representation of the four-state hidden Markov model is provided in Figure 7.19 that indicates the most likely operations for each state. Circular nodes indicate the hidden states, numbered 1 through 4. Rectangular nodes represent the most probable emissions from each hidden state (and are labeled with the appropriate operation name). The arrows connecting the nodes represent the probability of a transition (if the arrow connects two circular nodes) or of an emission (if the arrow connects a circular node to a rectangular node).

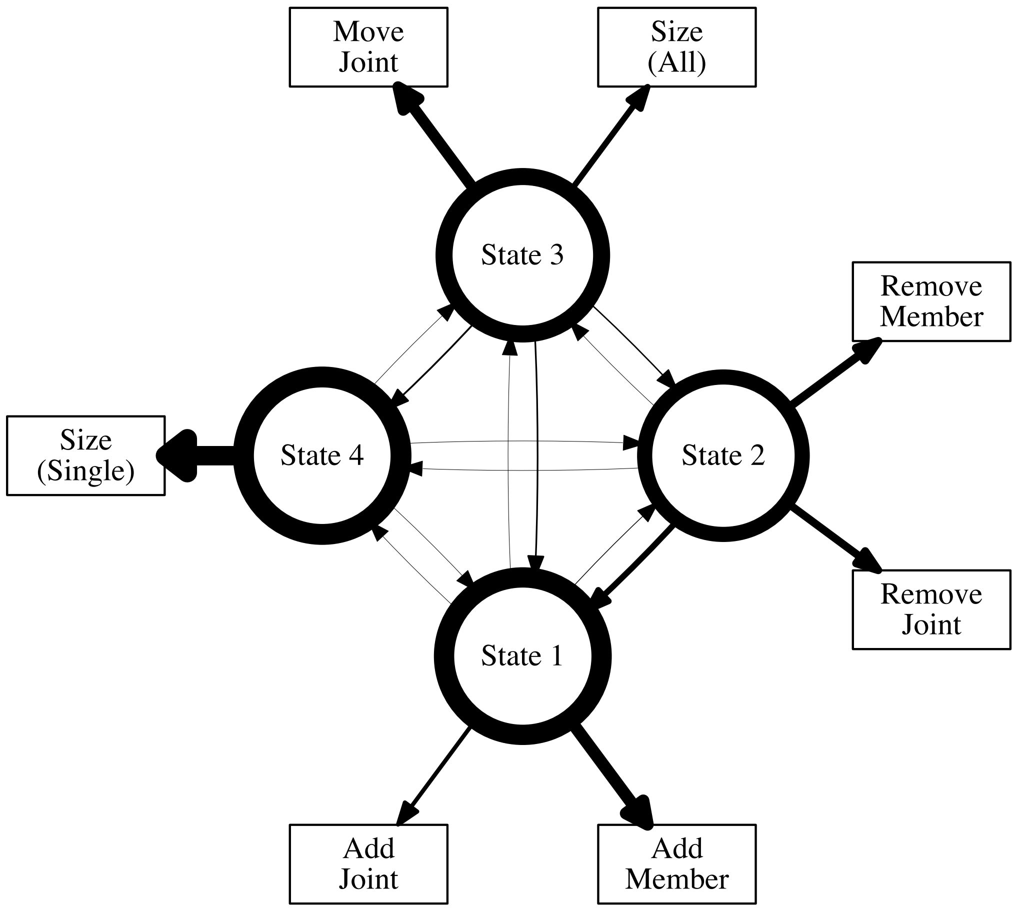


FIGURE 7.19. GRAPHICAL REPRESENTATION OF FOUR-STATE HIDDEN MARKOV MODEL LEARNED ON DATA FROM STUDY A. TRANSITION PROBABILITY IS INDICATED BY THE LINE THICKNESS OF THE CORRESPONDING ARROW. IN THE CASE OF SELF-TRANSITIONS, THE PROBABILITY IS INDICATED BY THE THICKNESS OF THE BORDER OF THE CORRESPONDING STATE.

Each of these hidden states corresponds to a specific design intention. In States 1 and 2 the designer is exclusively concerned with the topology of the truss. State 1 specifically corresponds to construction of a truss topology (through joint addition and member addition) while State 2 corresponds to destruction of parts of the truss topology (through joint and member removal). There is a 3% chance of transitioning from State 1 to State 2, but a 27% chance of transitioning from State 2 back to State 1. This indicates that the default topology mode of designers during the study was construction – transitions to the destructive state were rare and did not last long before a return to construction.

In States 3 and 4 the designer is exclusively concerned with the modification of truss parameters within a fixed topology. In State 3 this involves the movement of joints or the modification of the size of all structural members simultaneously. Both of these operations impact multiple structural elements at the same time, and this kind of coarse parametric optimization provides a good complement to the earlier topology optimization. A designer in State 3 has a 78% chance of remaining in that state. However, when they finally leave the state, they have a higher chance of transitioning back to one of the topology states (14%) than transitioning forward to the other parameter state (8%). Once a designer reaches State 4 they are highly likely to stay there, having only a 4% chance of leaving the state after each operation. In this state, designers apply a single operation that changes the size of a single member at a time. This impacts only one structural member at a time, and is thus indicative of very detailed design that might occur when a solution is nearly complete.

This four-state hidden Markov model provides a succinct representation of design activity as it relates to the designer's intentions: truss topology construction (State 1), truss topology trimming and refinement (State 2), coarse parameter optimization (State 3), and detailed parameter optimization (State 4).

7.6.3 Results for Study B

Figure **7.20** shows the results of training hidden Markov models with varying numbers of hidden states on data from Study B (the cooling system design problem). Models trained with 3 or fewer hidden states had testing log-likelihood values that were significantly lower than the most complex model (which had 9 hidden states), indicating that these models offered comparatively low accuracy. The first model that was not statistically different in terms of testing log-likelihood was the model trained with 4 hidden states. Therefore, that model is preferred in this application since it offers a balance between accuracy and simplicity.

This is the same number of hidden states selected for Study A. This is surprising given the fact that the design problems were in two significantly different domains (structural for Study A and thermal for study B). The number of operations used per participant to solve the two problems also differed by nearly an order of magnitude (400-500 operations/participant for Study A and 50 operations/participant in Study B).

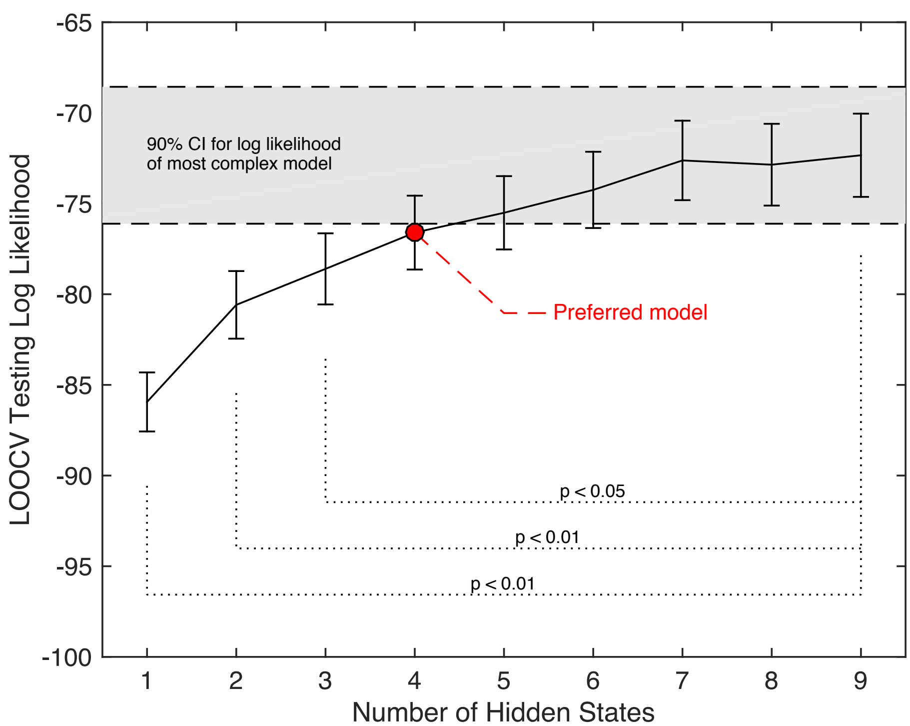


FIGURE 7.20. TESTING LOG-LIKELIHOOD OF MODELS WITH INCREASING NUMBER OF HIDDEN STATES LEARNED ON DATA FROM STUDY B. ERROR BARS SHOW ±1 S.E. AND P-VALUES INDICATE THE RESULTS OF ANOVA TESTS.

The transition and emission matrices associated with this four-state model are provided in Figure **7.21**. The color of each square in the matrices indicates the magnitude of the element, and the probability value is also denoted by the text within the square. In this model the transition matrix has some strong diagonal elements, much like Study A. However, unlike Study A, several off-diagonal elements also have large values, in some cases greater than the values of the diagonal elements. While all except one diagonal element is greater than 50%, none exceed 90%. In addition, one of the off-diagonal elements is larger than 50%, indicating a very common transition between states. Altogether, these observations show that designers in Study B frequently moved between strategic states, whereas designers in Study A tended to remain within a state for longer periods.

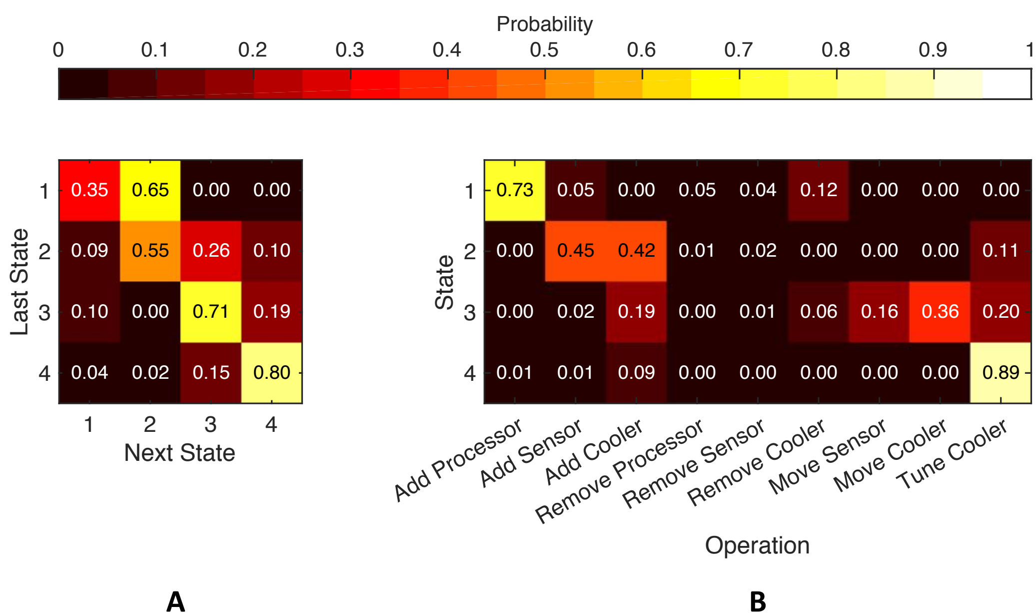


FIGURE 7.21**. PARAMETERS OF FOUR-STATE HIDDEN MARKOV MODEL BASED ON DATA FROM STUDY B, SHOWING (A) TRANSITION MATRIX AND (B) EMISSION MATRIX.**

A graph-based representation of the model is also provided in Figure 7.22 that indicates the most likely operations for each state. Circular nodes indicate the hidden states, numbered 1 through 4. Rectangular nodes represent the most probable emissions from each hidden state (and are labeled with the appropriate operation name). The arrows connecting the nodes represent the probability of a transition (if the arrow connects two circular nodes) or of an emission (if the arrow connects a circular node to a rectangular node).

As with Study A, the hidden states discovered for the cooling system design task also encode higher-level designer behavior, which may be interpreted as designer intent. State 1 corresponds to the creation of new subsystems – the most likely operation in the state is the addition of a processor. However, the probability of staying in this state is quite low (only 35%). Participants were far more likely to transition to State 2 in which the predominant operations were the addition of sensors and coolers. This shows that designers in Study B tended to create one subsystem and then elaborate it instead of instantiating multiple subsystems and elaborating them all simultaneously. From State 2, the probability of transitioning to State 3 is 26%. In State 3, a number of operations are employed – the movement of coolers and sensors, the addition of coolers, and the tuning of cooler parameters. State 4, in contrast, is dominated almost entirely by the tuning of cooler parameters.

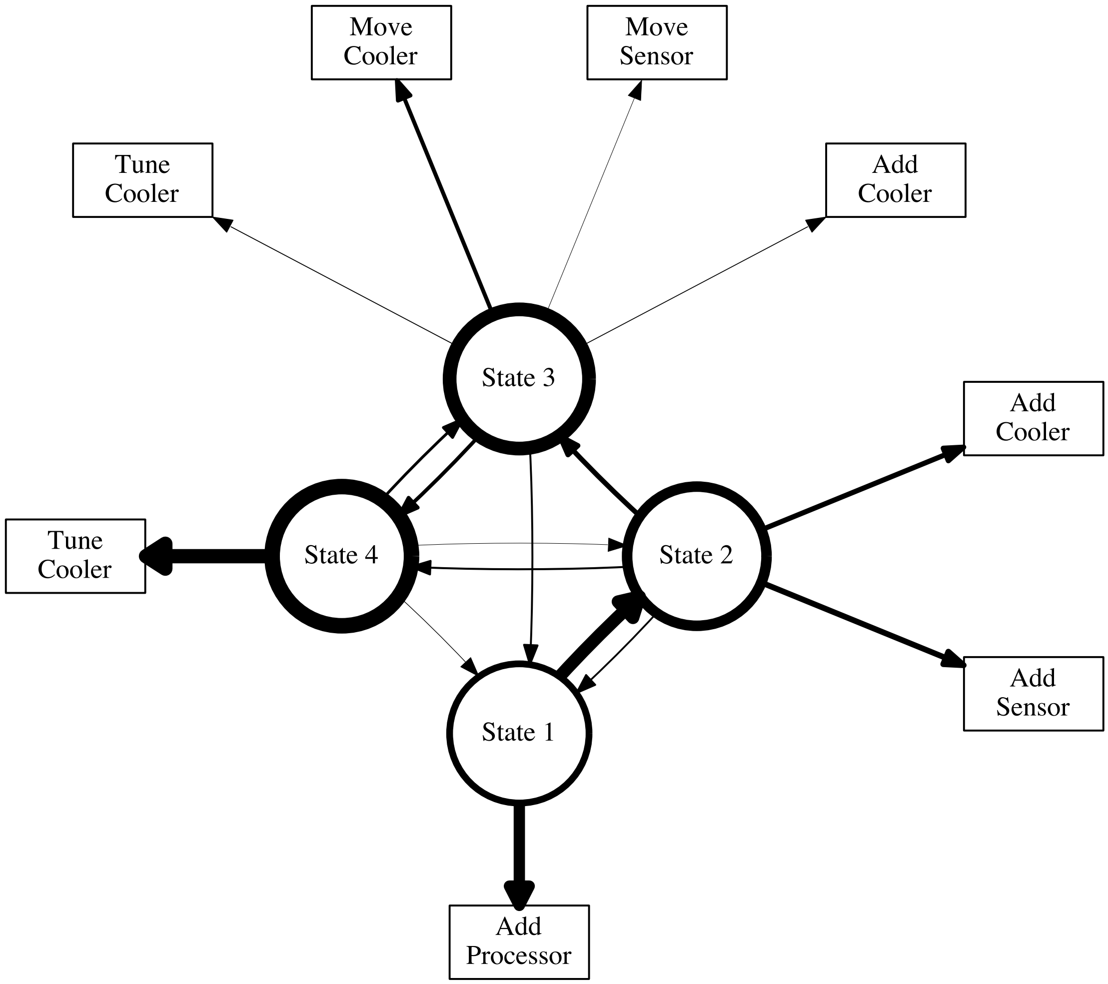


FIGURE 7.22. GRAPHICAL REPRESENTATION OF FOUR-STATE HIDDEN MARKOV MODEL LEARNED ON DATA FROM STUDY B. TRANSITION PROBABILITY IS INDICATED BY THE LINE THICKNESS OF THE CORRESPONDING ARROW. IN THE CASE OF SELF-TRANSITIONS, THE PROBABILITY IS INDICATED BY THE THICKNESS OF THE BORDER OF THE CORRESPONDING STATE.

Moreover, States 3 and 4 have a cyclic relationship to one another. The probability of transitioning from State 3 to State 4 is 19%, which is greater than the probabilities of transitioning to States 1 and 2 combined. The probability of transitioning from State 4 to State 3 is 15%, which is once again higher than the probabilities of transitioning to States 1 and 2 combined. This relationship indicates that once a designer reached either of these states he or she entered a cyclic tuning process in which system components were tuned, rearranged, and sometimes added. However, the addition (State 1) or expansion (State 2) of a new subsystem was relatively unlikely.

7.6.4 Discussion

This section searched for higher-level strategic sequences in the design sequence data through the use of hidden Markov models. Surprisingly, the models selected for both tasks contained four hidden states. A summary of the hidden Markov states identified in studies A and B is provided in Table 7.1. For both models, the first two states are purely focused on the topology of the potential solution. The third state is the only state in which spatial operations (physical relocation of solution components) are applied, although other operation types may also be employed. The fourth and final state focuses purely on optimizing the non-spatial parameters of the solution.

TABLE 7.1. SUMMARY OF STATES IDENTIFIED IN STUDY A AND STUDY B.

State	Study A		Study B	
	Operations	Type	Operations	Type
1	Add Joint Add Member	Topology Topology	Add Processor	Topology
2	Remove Joint Remove Member	Topology Topology	Add Cooler Add Sensor	Topology Topology
3	Move Joint Size (All)	Spatial Parameter Non-spatial Parameter	Add Cooler Move Sensor Move Cooler Tune Cooler	Topology Spatial Parameter Spatial Parameter Non-spatial Parameter
4	Size (Single)	Non-spatial Parameter	Tune Cooler	Non-spatial Parameter

Together, these results indicate that designers proceed through three strategic phases as they construct solutions to configuration problems. First, designers conceptualize the topology of their solution. After constructing a suitable topology, designers turn their attention to the spatial relationship between their solution concept and the constraints or requirements of the design space. The distinguishing characteristic of this stage is the employment of operations that change the location of design components – actions that are not employed in any other stages. This spatial negotiation stage may involve some necessary topology operations as well as the modification of non-spatial parameters that are associated with specific components. Only after performing this spatial negotiation do designers proceed to a final parameter optimization phase, in which they iterate over the non-spatial parameters associated with specific components in the solution in order to move towards a local minimum in the solution space.

7.7 Summary

This chapter examined operation sequencing in engineering design using a variety of statistical and computational tools. Three research questions were specifically addressed:

1. Do designers employ recognizable sequences of operations when solving a design problem, and if so, how might these sequences be characterized?
2. Is the employment of sequences of operations beneficial to designers?
3. Is the order in which operations occur indicative of higher-level strategies?

To address the first question, Markov chain models with increasing model order (representative of how much "memory" is assumed in the model) were trained on the data from two human studies. For both studies, the analysis indicated that the ordering of design actions might be treated accurately as a first-order Markov process. It should be noted that longer sequences of operations may still be extracted by traversing the graphs described by first-order Markov transition matrices.

To address the second question, computational simulations were performed to assess whether or not the ability to learn and employ operation sequences was beneficial to study participants. Several sets of simulations were conducted in which team of agents solved the design problems from Study A or Study B, either with the ability to learn sequences (encoded within a first-order Markov model) or without that ability (represented mathematically by non-sequential statistical model). It was shown that the ability to learn operation sequences significantly increased the quality of solutions in both design studies.

To address the third question of the existence of higher-level strategies, we returned to data from the two cognitive design studies, and hidden Markov models were used to extract higher-level sequences from the data. It was identified that data from both studies was represented well using hidden Markov models with four hidden states – two corresponding primarily to topology, one in which spatial modifications took place, and a fourth devoted almost completed to parameter tuning. This indicates that designers solve configuration problems by proceeding through three strategic phases. They first set a solution topology, then negotiate the spatial relation of the topology to the design situation, and finally tuning non-spatial parameters of specific components of the solution. This pattern is based on the commonalities of two disparate design studies, but it is unclear whether this pattern is generalizable to all configuration design problems. Examining this pattern of designers' behavior and intention in other configuration tasks is left to future work.

Together, these findings indicate that designers can benefit from recognizing and embracing beneficial sequences of operations while solving iterative design problems. This allows them to improve the quality of their solutions with greater efficiency, thus achieving the delivery of design solutions of higher final quality.

Chapter 8: Conclusion

If you want a happy ending, that depends, of course,
on where you stop your story.

Orson Welles

8.1 Overview

Teams are ubiquitous in a number of domains, and although they have the potential to search effectively for solutions they are also prone to a number of pitfalls. A greater understanding of teams is necessary to ensure that they can function optimally across a variety of tasks, but traditional team-based studies are limited by the cost required to run them, and the difficulty of inferring underlying cognitive processes. Many computational models have been developed to augment traditional studies, but these are either burdensome in their complexity or limited in their applicability. The following thesis was offered as a potential remedy for the above issues:

> Capturing characteristics of human teams within an accurate computational framework enables the derivation of a theory linking problem properties to optimized team characteristics, at a scope that would prove infeasible with empirical studies alone.

This thesis was first supported by the development of a core team-based simulation framework that captured several attributes of design teams as well as individual designers. The framework was then used to create a theoretical relationship between the properties of design problems and optimized team characteristics for solving those problems. In essence, this theory enabled the optimal design of design teams. This theory was subsequently validated in a limited cognitive study of human design teams. Two additional lines of research that were enabled by the simulation framework were also presented.

A cognitive study was presented in Chapter 2 and used to motivate the work in this dissertation. In the study, teams of engineering students were tasked with the design of a truss structure. Over the course of the study, participants contended with two drastic changes to their problem statement. Following the study, teams were gradated according to performance, and the top 5 teams (of 16 teams total) were designated as high-performing, while the bottom 5 teams were designated as low-performing. The primary result from this study was that high-performing teams displayed relatively low divergence, contrasting with several studies that indicate the benefit of divergent search. The superiority of this low-divergence approach might have been a result of the properties of the task itself, thus motivating an investigation into the relationship between design task properties, team characteristics, and team performance.

In Chapter 3, the Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework was developed with the intention of providing a resource-efficient testbed with direct applicability to engineering design problems while maintaining relative model simplicity. Simulated annealing has previously been used as a model of individual problem-solving and design activity, so every agent in CISAT was given control over its own simulated annealing algorithm. These agents were then situated within a multi-agent structure to allow them to interact. Next, a variety of characteristics of teams and individuals were selected from the engineering, design, and psychology literature and layered on top of the multi-agent simulated annealing framework to facilitate a more robust depiction of engineering design teams. The ability of CISAT to accurately mimic the process and performance of engineering design teams was verified by using it to directly simulate the results of a prior cognitive study. A comparison of the CISAT simulated results to those of the original cognitive study revealed a high degree of correlation between the two. Next, CISAT was used to explore the particular characteristics that were most beneficial to teams in solving this task. This analysis indicated that the presence of moderate self-bias and infrequent interaction were crucial to enabling team success in the truss design task.

The efficiency of CISAT was exploited in Chapter 4 to explore the relationship between several design problem properties and the selection of optimal team characteristics. Three design problem properties were first defined: the local structure of the design space, the global structure, and the alignment between objective functions. These properties were then used to characterize a number of configuration design problems. The CISAT modeling framework was then used to simulate the performance of engineering design teams with a broad variety of team characteristics on each of the predefined design problems, resulting in the determination of the team characteristics leading to the best performance on each problem. Regression analysis was applied to create equations linking the optimal team characteristics to the properties describing the design problems. These equations essentially enabled the prediction of optimal design team characteristics (team size and interaction frequency) based on design problem properties. An examination of the equations indicated that the selection of the optimal number of individuals in a team is described by a complicated relationship, depending greatly on the design space properties as well as the interactions between them. The optimal interaction frequency can be predicted with high accuracy based on the linear effects of the design space properties without the need to consider interaction effects.

The equations produced in Chapter 4 were derived entirely from CISAT computational simulations. In Chapter 5, a cognitive study was conducted in order to offer a limited validation of the computationally-derived theory. A configuration problem of moderate size was first defined, and then its properties were measured. The values of these properties were then used to predict the optimal interaction frequency for solving the problem. Surprisingly, the predicted optimal frequency was zero, indicating that individuals should work independently to solve the problem! This prediction was tested in a cognitive study of human design teams. The study demonstrated that non-interacting teams performed better than frequently interacting teams, thus agreeing with the prediction and provided limited validation of the theory. The results of a survey of the participants revealed that although non-interacting teams provided the best solutions and experienced the least difficulty, they also had the lowest satisfaction with their performance. This misalignment between performance and personal satisfaction is an important result that has potential implications for long-term maintenance of morale in teams and organizations.

At its core, CISAT makes use of simulated annealing, a well-known optimization algorithm. Because of this association, the possibility of deriving a novel optimization algorithm directly from the CISAT framework was explored in Chapter 6. Thus, the Heterogeneous Simulated Annealing Teams (HSAT) algorithm was born by stripping away all characteristics from CISAT but those that were essential to providing a depiction of team-based activity. The three characteristics retained were multi-agency (the composition of a team from individuals), quality-informed solution sharing (enabling collaboration between agents), and locally sensitive search (enabling the development of heterogeneous, agent-specific strategies within the team). The performance of HSAT was then benchmarked by using it to minimize several highly-multimodal, high-dimensional functions and the performance of HSAT was compared to several other algorithms within the simulated annealing class. HSAT was the only algorithm that offered consistently high performance on every benchmarking function. On the Ackley function, this level of performance was attributed to the ability to engage in locally sensitive search; on the Griewank function, high performance was attributed to the interaction between agents; and on the Rastrigin function, both of these characteristics were necessary to produce high performance. Thus, the specific set of CISAT characteristics that were retained in HSAT provided flexibility resulting in robust performance across a variety of function topographies.

It was hypothesized in Chapter 3 that the specific order in which design operations were performed contributed substantially to team performance. This and other aspects of design sequencing were the focus of Chapter 7. First, operational data from the cognitive studies in Chapters 2 and 5 was analyzed using Markov chains to determine whether or not designers employed recognizable sequences, and if so, how much inherent memory was necessary to describe the sequences. It was determined that the ordering of design actions in both studies could be treated accurately as a first-order Markov process, implying that strong sequences existed in the data but that lengthy memory of operations was unnecessary to encode them. Second, computational simulations were performed to assess whether or not the sequencing of design operations resulted in solutions of higher quality. CISAT was used to simulate the performance of teams with and without the ability to learn and employ sequences of operations. It was shown that sequence-learning did have a positive effect on terminal solution quality. Third, hidden Markov models were trained on the operational data from the two cognitive studies, resulting in a statistical model of the strategic approaches employed by designers. It was identified that data from both studies was represented well using hidden Markov models with four hidden states – two corresponding primarily to topology, one in which spatial parameter modifications took place, and a fourth devoted almost completely to non-spatial parameter tuning. The degree of similarity between the patterns employed by designers in both studies provides evidence of a possibly general pattern of design for configuration problems.

8.2 Contributions

This dissertation made the following contributions to the engineering design community:

1. *The creation of an approach for simulating human design teams by capturing characteristics of both individual designers and teams within a multi-agent simulated annealing framework.*

   Existing computational models of human teams tend to be either exceedingly complex [27,28] or are not capable of solving real-world problems and thus cannot be readily validated [29–32]. Simulated annealing has been used previously as a model of individual problem-solving [33] and individual design [34]. The CISAT framework introduced here is the first use of multi-agent simulated annealing to model human design teams, and this model achieves relatively low complexity while still being directly applicable to engineering design problems.

2. *The demonstration that individual bias towards one's own solutions may promote the formation of weak ties between design team members, which may in turn enable teams to produce higher quality solutions.*

   It is generally assumed that self-bias is detrimental in teams [77–80]. However, self-bias can form weak ties, which are known to have a beneficial effect on information flow in networks [100,101]. Simulations in CISAT demonstrate that self-bias may lead to better solutions for some engineering design problems.

3. *The derivation of a theory that makes it possible to construct better design teams by enabling the selection of the best values for team characteristics (such as team size and interaction frequency) based on measurements of design problem properties.*

   The selection of both optimal team size [22,113] and optimal interaction frequency [21,123,124] have been studied by others. However, available relationships are based on disparate sets of problem properties. Further, they have not been developed for problem types that are relevant to engineering design. This work developed equations for both team size and interaction frequency based on a common set of problem properties that are specifically relevant to engineering design.

4. *The subsequent limited validation of the above theory through a cognitive study that challenged common assumptions about team versus individual effectiveness.*

   It is commonly assumed that interacting teams are superior to non-interacting teams [13]. Using the predictive theory from the previous contribution, it was shown that it is possible to predict design problems for which interacting teams are in fact inferior to non-interacting teams.

5. *The creation of a novel and robust optimization algorithm derived directly from a computational model of engineering design teams based on human characteristics.*

   Other multi-agent simulated annealing algorithms have been developed [180,181], but all use a single temperature schedule that is enforced on all agents. As a result of its origin from a model of human designers, the HSAT algorithm grants each agent independent control over a unique adaptive annealing schedule. It was demonstrated that the democratization of the annealing schedule in this fashion can result in more efficient optimization.

6. *The demonstration that human designers use strong sequences of operations, and that such sequencing can be treated accurately as a first-order Markov process.*

   Other work has used first-order [214–216] and second-order [217] Markov chains to describe human design sequences, but it is unclear what model order is necessary to accurately depict human activity. This work trained Markov chains of varying order on human-generated data to demonstrate that first-order Markov chains can accurately encode human operation sequencing.

7. *The demonstration that the ability to learn and employ sequences during design is beneficial to human designers.*

   Because sequence learning can occur as both an implicit process [25,26] and an explicit process [23,24], it is extremely difficult (if not impossible) to directly test the effects of sequence learning using traditional human studies.

CISAT was used to computationally compare the performance of teams with and without the ability to learn sequences, demonstrating that the ability is beneficial to designers.

8. *The first application of hidden Markov models to study the activity of human designers, resulting in the identification of a new pattern of design that applies to configuration problems.*

   The identification of procedural patterns of designer activity typically involves protocol encoding (for examples see [75,77,216]) which is tedious and also introduces subjectivity. In this work, hidden Markov models were used to automate the extraction of a cognitive pattern for configuration design problems.

8.3 Areas for Future Work

Future research that builds upon the work presented in this dissertation can be broadly split into two tracks: modeling and studying human teams; and developing team- and human-inspired design algorithms.

8.3.1 Modeling and Studying Human Teams

The focus of this dissertation was on teams that are homogeneous with respect to both skillset and experience as well as communication hierarchy. A promising avenue of future work would be to extend the CISAT modeling framework to account for heterogeneity of these attributes. Enabling CISAT agents to be instantiated with various and unique skillsets (treated computationally as the ability to modify only certain variables, or specific operational heuristics) would make it possible to model the process and performance of cross-functional teams. This could allow CISAT to explore issues that are of particular importance to these teams, such as the process of recognizing and negotiating perceptual gaps [238,239], as well as the balance between cooperation and competition within cross-functional teams [240]. Almost half of cross-functional team members experience cooperation problems within their team [241]. In addition, a high percentage of cross-functional teams may fail basic performance criteria like adhering to a schedule, maintaining a budget, or meeting customer expectations [242]. Computational modeling efforts could develop best practices to help solve these problems. Further, enabling CISAT to model heterogeneous communication structures would provide the ability to simulate multi-team organizations. This could be combined with skillset heterogeneity to model hierarchical teams in which sub-teams oversee the design of specific systems (e.g., NASA's Advanced Projects Design Team). Several models of negotiation in engineering design have been published [243,244] that could form the backbone of an inter-team communicative structure in CISAT.

The computationally derived theory outlined in Chapter 4 was validated through a single study of human design teams in Chapter 5, and results showed strong support for the theoretical prediction. Regardless, additional studies should be performed to provide a broader base of validation and to better assess the accuracy of the theory's predictions. The theory also posits several strong linear relationships between design problem properties and team characteristics, and these should be examined as well. For instance, objective function alignment was shown to have a strong positive correlation with interaction frequency. In other words, the theory predicts that problems that are unlikely to require difficult tradeoffs between objectives tend to be most effectively solved by a frequently interacting team. This computationally demonstrated relationship and others should be investigated experimentally.

Hidden Markov models were used in Chapter 7 to extract a new pattern of design applying to human behavior when solving configuration problems. This pattern indicates three stages of action through which designers progress nonlinearly: topological definition, spatial negotiation, and detailed tuning. The two design problems analyzed in the chapter were different in terms of domain (thermal versus structural) and the number of operations used (differing here by an order of magnitude). This offers a strong indication that the pattern may be general, but future work should investigate the generalizability with more rigorous validation on other configuration design problems.

8.3.2 Developing Team- and Human-Inspired Design Algorithms

This dissertation also stands to inform future research and development of design algorithms. In Chapter 6, the HSAT algorithm was derived *directly* from CISAT, simply by the removal of characteristics that, while necessary to create human-like behavior, would have perhaps proven cumbersome during optimization. This demonstrates a prototypical process by which phenomena observed in human designers can be transferred and utilized in design algorithms. The key is that CISAT is capable of creating real solutions to engineering design problems. If cognitive phenomena are implemented in a model like CISAT while they are being studied, then the creation of novel team-inspired design algorithms is a straightforward process. In this way, explorations into cognitive and social aspects of human design can quickly inform the creation of new and improved design tools.

In Chapter 4 of this dissertation, a set of simple equations were created to enable the prediction of the optimal team characteristics for solving a problem based on measurements of the properties of the problem. The same methodology used in that chapter could be applied to develop heuristics for selecting the best parameters to use in optimization algorithms. For instance, evolutionary algorithms often require a variety of parameters to be tuned, including population size, mutation rate, crossover rate, and potentially many others [245]. The methodology in Chapter 4 could be applied to provide efficient and simple equations to link measurable properties of the objective functions to the best parameters to use in the evolutionary algorithm. It may even be possible to use the same problem properties used in this work: objective alignment (based on the rank correlation between objective functions), local structure (based on the fractal dimension of the space), and global structure (based on the Hurst exponent of the space). These properties have the benefit that they can be computed with a simple random walk through the design space. This could lead to the development of simple heuristics for tuning a wide variety of algorithms, increasing their user-friendliness and leading to more widespread adoption of advanced optimization in industry.

The sequence-based modeling approaches used in Chapter 7 are generative [246], meaning that they encode the training data in such a way that they are capable of creating synthetic data. Such models could be implemented in design or optimization algorithms to generate synthetic sequences of beneficial operations, enabling computational agents to exploit sequence-based expedients in the design space. These constructs could be trained a priori if data is available, or trained during solving as an algorithm actively explores potential solutions to the design problem. Markov chains are best suited to this purpose, since they can be operationalized using a simple reinforcement approach similar to that used to enable CISAT agents to learn sequences. Hidden Markov models could be useful as well, particularly for their ability to encode a representation of design intention and strategy. The main barrier to their potential use is the high computational cost of training a hidden Markov model. Therefore, it is likely that hidden

Markov models would need to be trained prior to use in an algorithm, rather than being trained during solving.

8.4 Coda

This work demonstrated how characteristics of individual designers and design teams could be captured and accurately reproduced within a computational model. This model was utilized to conduct research at a previously impractical scale, performing the equivalent of hundreds of thousands of hours of human cognitive studies. This led to the creation of a computationally derived theory linking problem properties to optimized team characteristics, and the limited validation of the theory through a cognitive study of human teams. The core computational model was developed with versatility in mind, enabling two additional lines of inquiry that resulted in a new team-inspired optimization algorithm and a greater understanding of the ordering of operations in design tasks. This work and future extensions of it have the potential to inform a deeper and more holistic understanding of the search process as it relates not only to humans, but also to computational agents.

References

[1] Devine, D. J., Clayton, L. D., Philips, J. L., Dunford, B. B., and Melner, S. B., 1999, “Teams in Organizations: Prevalence, Characteristics, and Effectiveness,” Small Group Research, **30**(6), pp. 678–711.

[2] Kozlowski, S. W. J., and Ilgen, D. R., 2006, “Enhancing the effectiveness of work groups and teams,” Psychological Science, **7**(3), pp. 77–124.

[3] Paulus, P. B., Dzindolet, M. T., and Kohn, N., 2011, “Collaborative creativity-Group creativity and team innovation,” Handbook of organizational creativity, M.D. Mumford, ed., Elsevier, New York, pp. 327–357.

[4] Reiter-Palmon, R., Herman, A. E., and Yammarino, F. J., 2008, “Creativity and cognitive processes: Multi-level linkages between individual and team cognition,” Multi-Level Issues in Creativity and Innovation (Research in Multi Level Issues, Volume 7), M.D. Mumford, S.T. Hunter, and K.E. Bedell-Avers, eds., pp. 203–267.

[5] Wuchty, S., Jones, B. F., and Uzzi, B., 2007, “The increasing dominance of teams in production of knowledge.,” Science, **316**(5827), pp. 1036–9.

[6] Cannon-Bowers, J. A., Salas, E., and Converse, S., 1993, “Shared mental models in expert team decision making,” Individual and Group Decision Making: Current Issues, N.J. Castellan, ed., pp. 221–246.

[7] Dyer, D. J., 1984, “Team research and team training: A state-of-the-art review,” Human Factors Review: 1984, F. Muckler, ed., Human Factors Society, Santa Monica, pp. 285–323.

[8] Orasanu, J. M., and Salas, E., 1993, “Team decision making in complex environments,” Decision making in action: models and methods, G. Klein, J. Orasanu, R. Calderwood, and C. Zsambok, eds., Ablex Publishers, Norwood, NJ, pp. 327–345.

[9] Morgan, B., Glickman, A., Woodard, E., Blaiwes, A., and Salas, E., 1986, Measurement of Team Behaviors in a Navy Environment.

[10] Miller, G. A., 1956, “The magical number seven, plus or minus two: some limits on our capacity for processing information.,” Psychological Review, **63**(2), pp. 81–97.

[11] Simon, H. A., 1974, “How Big Is a Chunk?,” Science, **183**(4124), pp. 482–488.

[12] Kozlowski, S. W. J., Gully, S. M., Nason, E. R., and Smith, E. M., 1999, “Developing adaptive teams: A theory of compilation and performance across levels and time,” The changing nature of work performance: Implications for staffing, personnel actions, and

development, D.R. Ilgen, and E.D. Pulakos, eds., Jossey-Bass, San Francisco, CA, pp. 240–292.

[13] Katzenbach, J. R., and Smith, D. K., 1993, The Wisdom of Teams: Creating the High-Performance Organization.

[14] Mumford, M. D., Feldman, J. M., Hein, M. B., and Nagao, D. J., 2001, “Tradeoffs between ideas and structure: Individual versus group performance in creative problem solving,” The Journal of Creative Behavior, **35**(1), pp. 1–23.

[15] Diehl, M., and Stroebe, W., 1987, “Productivity loss in brainstorming groups: Toward the solution of a riddle.,” Journal of Personality and Social Psychology, **53**(3), pp. 497–509.

[16] Mullen, B., Johnson, C., and Salas, E., 1991, “Productivity Loss in Brainstorming Groups: A Meta-Analytic Integration,” Basic and Applied Social Psychology, **12**(1), pp. 3–23.

[17] Janis, I., 1971, “Groupthink,” Psychology Today, **5**(6), pp. 43–46.

[18] Jones, P. E., and Roelofsma, P., 2000, “The potential for social contextual and group biases in team decision-making: biases, conditions and psychological,” Ergonomics, **43**(8), pp. 1129–1152.

[19] Stoner, J. A. ., 1968, “Risky and cautious shifts in group decisions: The influence of widely held values,” Journal of Experimental Social Psychology, **4**(4), pp. 442–459.

[20] Harkins, S. G., Latané, B., and Williams, K., 1980, “Social loafing: Allocating effort or taking it easy?,” Journal of Experimental Social Psychology, **16**(5), pp. 457–465.

[21] Kennedy, D. M., McComb, S. A., and Vozdolska, R. R., 2011, “An investigation of project complexity’s influence on team communication using Monte Carlo simulation,” Journal of Engineering and Technology Management, **28**(3), pp. 109–127.

[22] Heričko, M., Živkovič, A., and Rozman, I., 2008, “An approach to optimizing software development team size,” Information Processing Letters, **108**(3), pp. 101–106.

[23] Perruchet, P., and Amorim, M.-A., 1992, “Conscious Knowledge and Changes in Perfomance in Sequence Learning: Evidence Against Dissociation,” Journal of Experimental Psychology: Learning, Memory, and Cognition, **18**(4), pp. 785–800.

[24] Willingham, D. B., Nissen, M. J. J., and Bullemer, P., 1989, “On the development of procedural knowledge,” Journal of Experimental Psychology: Learning, Memory, and Cognition, **15**(6), pp. 1047–1060.

[25] Nissen, M. J., and Bullemer, P., 1987, “Attentional requirements of learning: Evidence from performance measures,” Cognitive Psychology, **19**(1), pp. 1–32.

[26] Reed, J., and Johnson, P., 1994, “Assessing implicit learning with indirect tests:

Determining what is learned about sequence structure.," Journal of Experimental Psychology: Learning, Memory, and Cognition, **20**(3), pp. 585–594.

[27] Jin, Y., and Levit, R., 1996, "The Virtual Design Team: A Computational Model of Project Organization," Computational & Mathematical organization Theory, **2**(3), pp. 171–196.

[28] Olson, J., Cagan, J., and Kotovsky, K., 2009, "Unlocking Organizational Potential: A Computational Platform for Investigating Structural Interdependence in Design," Journal of Mechanical Design, **131**, pp. 031001–1–13.

[29] Dionne, S. D., Sayama, H., Hao, C., and James, B., 2010, "The role of leadership in shared mental model convergence and team performance improvement : An agent-based computational model," The Leadership Quarterly, **21**(6), pp. 1035–1049.

[30] Singh, V., Dong, A., and Gero, J. S., 2012, "Computational studies to understand the role of social learning in team familiarity and its effects on team performance," CoDesign, **8**(1), pp. 25–41.

[31] Sayama, H., Farrell, D. L., and Dionne, S. D., 2011, "The Effects of Mental Model Formation on Group Decision Making: An Agent-Based Simulation," Complexity, **16**(3), pp. 49–57.

[32] Singh, V., Dong, A., and Gero, J. S., 2013, "Social learning in design teams: The importance of direct and indirect communications," Artificial Intelligence in Engineering, Design, Analysis and Manufacturing, **27**(2), pp. 167–182.

[33] Cagan, J., and Kotovsky, K., 1997, "Simulated annealing and the generation of the objective function: a model of learning during problem solving," Computational Intelligence, **13**(4), pp. 534–581.

[34] Yu, B. Y., Honda, T., Sharqawy, M., and Yang, M., 2016, "Human behavior and domain knowledge in parameter design of complex systems," Design Studies, pp. 1–26.

[35] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., 1983, "Optimization by simulated annealing," Science, **220**(4598), pp. 671–680.

[36] McComb, C., Cagan, J., and Kotovsky, K., 2015, "Rolling with the punches: An examination of team performance in a design task subject to drastic changes," Design Studies, **36**(1), pp. 99–121.

[37] Jansson, D., and Smith, S. M., 1991, "Design Fixation," Design Studies, **12**(1), pp. 3–11.

[38] Kotovsky, K., and Fallside, D., 1989, "Representation and Transfer in Problem Solving," Complex Information Processing, D. Klahr, and K. Kotovsky, eds., Lawrence Erlbaum Associates, pp. 69–108.

[39] LePine, J., Colquitt, J., and Erez, A., 2000, “Adaptability to changing task contexts: Effects of general cognitive ability, conscientiousness, and openness to experience,” Personnel Psychology, **53**(3), pp. 563–593.

[40] LePine, J., 2003, “Team adaptation and postchange performance: Effects of team composition in terms of members’ cognitive ability and personality.,” Journal of Applied Psychology, **88**(1), pp. 27–39.

[41] LePine, J., 2005, “Adaptation of teams in response to unforeseen change: effects of goal difficulty and team composition in terms of cognitive ability and goal orientation.,” The Journal of Applied Psychology, **90**(6), pp. 1153–1167.

[42] Cross, N., 2004, “Expertise in design: an overview,” Design Studies, **25**(5), pp. 427–441.

[43] Ericsson, K. A., Krampe, R. T., and Tesch-romer, C., 1993, “The Role of Deliberate Practice in the Acquisition of Expert Performance,” Psychological Review, **100**(3), pp. 363–406.

[44] Fu, K., Cagan, J., and Kotovsky, K., 2010, “Design Team Convergence: The Influence of Example Solution Quality,” Journal of Mechanical Design, **132**(11), p. 111005.

[45] Dong, A., Hill, A. W., and Agogino, A. M., 2004, “A Document Analysis Method for Characterizing Design Team Performance,” Journal of Mechanical Design, **126**(3), pp. 378–385.

[46] Dong, A., and Agogino, A. M., 1997, “Text analysis for constructing design representations,” Artificial Intelligence in Engineering, **11**(2), pp. 65–75.

[47] Hibbeler, R., 2008, Structural Analysis, Prentice Hall.

[48] Rahami, H., 2007, “Truss Analysis,” Mathworks File Exchange Software.

[49] Arora, S., 1998, “The approximability of NP-hard problems,” Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC ’98, pp. 337–348.

[50] Wood, M., Chen, P., Fu, K., Cagan, J., and Kotovsky, K., 2014, “The Role of Design Team Interaction Structure on Individual and Shared Mental Models,” Design Computing and Cognition ’12, J.S. Gero, ed., Springer Netherlands, Dordrecht, pp. 209–226.

[51] Wood, M., 2013, “Problem Representation and Team Mental Model Development in Individual and Team Problem-Solving Performance,” Carnegie Mellon University.

[52] Vandewalle, D., 1997, “Development and Validation of a Work Domain Goal Orientation Instrument,” Educational and Psychological Measurement, **57**(6), pp. 995–1015.

[53] Kaplan, A., and Maehr, M. L., 2006, “The Contributions and Prospects of Goal Orientation Theory,” Educational Psychology Review, **19**(2), pp. 141–184.

[54] Kirschner, P., 2002, “Cognitive load theory: implications of cognitive load theory on the design of learning,” Learning and Instruction, **12**(1), pp. 1–10.

[55] Mayer, R. E., and Moreno, R., 2003, “Nine Ways to Reduce Cognitive Load in Multimedia Learning,” Educational Psychologist, **38**(1), pp. 43–52.

[56] Kotovsky, K., Hayes, J. ., and Simon, H. A., 1985, “Why are some problems hard? Evidence from Tower of Hanoi,” Cognitive Psychology, **17**(2), pp. 248–294.

[57] Kotovsky, K., and Simon, H. A., 1990, “What makes some problems really hard: Explorations in the problem space of difficulty,” Cognitive Psychology, **22**(2), pp. 143–183.

[58] Chi, M., Glaser, R., and Farr, M., 1988, The Nature of Expertise, Psychology Press.

[59] Chi, M., Feltovich, P. J., and Glaser, R., 1981, “Categorization and Representation of Physics Problems by Experts and Novices,” Cognitive Science, **5**(2), pp. 121–152.

[60] Fricke, G., 1996, “Successful individual approaches in engineering design,” Research in Engineering Design, **8**(3), pp. 151–165.

[61] Badke-Schaub, P., Neumann, A., Lauche, K., and Mohammed, S., 2007, “Mental models in design teams: a valid approach to performance in design collaboration?,” CoDesign, **3**(1), pp. 5–20.

[62] Bierhals, R., Schuster, I., Kohler, P., and Badke-Schaub, P., 2007, “Shared mental models—linking team cognition and performance,” CoDesign, **3**(1), pp. 75–94.

[63] Kruglanski, A. W., Webster, D. M., and Klem, A., 1993, “Motivated resistance and openness to persuasion in the presence or absence of prior information.,” Journal of Personality and Social Psychology, **65**(5), pp. 861–876.

[64] Kerr, N. L., and Tindale, R. S., 2004, “Group Performance and Decision Making,” Annual Review of Psychology, **55**(1), pp. 623–655.

[65] Osborn, A. F., 1957, Applied imagination., Scriber, New York.

[66] Brown, V., and Paulus, P. B., 2002, “Making Group Brainstorming More Effective: Recommendations from an associative memory perspective,” Current Directions in Psychological Science, **11**(6), pp. 208–212.

[67] Newell, A., and Simon, H. A., 1972, Human Problem Solving, Prentice Hall.

[68] Langley, P., 1985, “Learning to Search: From Weak Methods to Domain-Specific Heuristics,” Cognitive Science, **9**(2), pp. 217–260.

[69] Ball, L. J., and Ormerod, T. C., 1995, “Structured and opportunistic processing in design:

a critical discussion," International Journal of Human-Computer Studies, **43**(1), pp. 131–151.

[70] Simon, H. A., 1956, "Rational Choice and the Structure of the Environment," Psychological Review, **63**(2), pp. 129–138.

[71] Cooper, R., and Slagmulder, R., 1997, Target Costing and Value Engineering, Productivity Press, Portland, OR.

[72] Feil, P., Yook, K., and Kim, I., 2004, "Japanese target costing: a historical perspective," International Journal of Strategic Cost Management, pp. 10–19.

[73] Kurtzberg, T. R., and Amabile, T. M., 2001, "From Guilford to Creative Synergy: Opening the Black Box of Team-Level Creativity," Creativity Research Journal, **13**(3-4), pp. 285–294.

[74] Sio, U. N., Kotovsky, K., and Cagan, J., 2014, "Analyzing the Effect of Team Structure on Team Performance: An Experimental and Computational Approach," 36th Annual Conference of the Cognitive Science Society, P. Bello, M. Guarini, M. McShane, and B. Scassellati, eds., Cognitive Science Society, Austin, TX, pp. 1437–1442.

[75] Stempfle, J., and Badke-schaub, P., 2002, "Thinking in design teams - an analysis of team communication," Design Studies, **23**(5), pp. 473–496.

[76] Dieter, G., and Schmidt, L., 2009, Engineering Design, McGraw-Hill.

[77] Ullman, D. G., Dietterich, T. G., and Stauffer, L. A., 1988, "A model of the mechanical design process based on empirical data," Artificial Intelligence in Engineering, Design, Analysis and Manufacturing, **2**(1), pp. 33–52.

[78] Ball, L. J., Evans, J. S. B., and Dennis, I., 1994, "Cognitive Processes in Engineering Design: a longitudinal study," Ergonomics, **37**(11), pp. 1753–1786.

[79] Nikander, J. B., Liikkanen, L. A., and Laakso, M., 2014, "The preference effect in design concept evaluation," Design Studies, **35**(5), pp. 473–499.

[80] Toh, C. A., Strohmetz, A. A., and Miller, S. R., 2016, "The effects of gender and idea goodness on ownership bias in engineering design education," Journal of Mechanical Design.

[81] Fan, X., and Yen, J., 2004, "Modeling and simulating human teamwork behaviors using intelligent agents," Physics of Life Reviews, **1**(3), pp. 173–201.

[82] Takai, S., 2015, "A Multidisciplinary Framework to Model Complex Team-Based Product Development," Volume 4: 20th Design for Manufacturing and the Life Cycle Conference; 9th International Conference on Micro- and Nanosystems, ASME, p. V004T05A038.

[83] Crowder, R. M., Robinson, M. A., Hughes, H. P. N., and Sim, Y.-W., 2012, "The Development of an Agent-Based Modeling Framework for Simulating Engineering Team Work," IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, **42**(6), pp. 1425–1439.

[84] Dong, A., Garbuio, M., and Lovallo, D., 2016, "Generative sensing in design evaluation," Design Studies, pp. 1–24.

[85] Egan, P. F., Cagan, J., Schunn, C., and LeDuc, P. R., 2014, "Cognitive-Based Search Strategies for Complex Bio-Nanotechnology Design Derived Through Symbiotic Human and Agent-Based Approaches," Volume 7: 2nd Biennial International Conference on Dynamics for Design; 26th International Conference on Design Theory and Methodology, ASME, p. V007T07A024.

[86] Franklin, S., and Graesser, A., 1997, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents," Intelligent Agents III: Agent theories, architectures, and languages., pp. 21–35.

[87] Kumar, M., Campbell, M. I., Königseder, C., and Shea, K., 2014, "Rule Based Stochastic Tree Search," Design Computing and Cognition '12, J.S. Gero, ed., Springer Netherlands, Dordrecht, pp. 571–587.

[88] Campbell, M. I., Rai, R., and Kurtoglu, T., 2012, "A Stochastic Tree-Search Algorithm for Generative Grammars," Journal of Computing and Information Science in Engineering, **12**(3), p. 031006.

[89] Hustin, S., 1988, "TIM, A New Standard Cell Placement Program Based on the Simulated Annealing Algorithm," University of California, Berkeley.

[90] Shea, K., and Cagan, J., 1998, "Generating Structural Essays from Languages of Discrete Structures," Artificial Intelligence in Design '98, pp. 365–384.

[91] Triki, E., Collette, Y., and Siarry, P., 2005, "A theoretical study on the behavior of simulated annealing leading to a new cooling schedule," European Journal of Operational Research, **166**(1), pp. 77–92.

[92] Shea, K., and Cagan, J., 1999, "Languages and semantics of grammatical discrete structures," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **13**, pp. 241–251.

[93] Suppapitnarm, A., Parks, G. T., Shea, K., and Clarkson, P. J., 2004, "Conceptual design of bicycle frames by multiobjective shape annealing," Engineering Optimization, **36**(2), pp. 165–188.

[94] Konigseder, C., and Shea, K., 2016, "Comparing Strategies for Topologic and Parametric

Rule Application in Automated Computational Design Synthesis," Journal of Mechanical Design, Transactions of the ASME, **138**(1), pp. 1–12.

[95] Pretz, J. E., Naples, A. J., and Sternberg, R. J., 2003, "Recognizing, Defining, and Representing Problems," The Psychology of Problem Solving, J.E. Davidson, and R.J. Sternberg, eds., Cambridge University Press.

[96] Fritz, C., Morris, P., and Richler, J., 2012, "Effect size estimates:current use, calculations, and interpretation," **141**(1), pp. 2–18.

[97] Hess, M., and Kromrey, J., 2004, "Robust confidence intervals for effect sizes: A comparative study of Cohen'sd and Cliff's delta under non-normality and heterogeneous variances," Annual Meeting of the American Educational Research Association, pp. 1–30.

[98] Leech, N. L., and Onwuegbuzie, A. J., 2002, "A call for greater use of nonparametric statistics," Annual Meeting of the MidSouth Educational Research Association, pp. 1–24.

[99] Cliff, N., 1996, "Answering Ordinal Questions with Ordinal Data Using Ordinal Statistics," Multivariate Behavioral Research, **31**(3), pp. 331–350.

[100] Granovetter, M. S., 1973, "The Strength of Weak Ties," The American Journal of Sociology, **78**(6), pp. 1360–1380.

[101] Granovetter, M., 1983, "The Strength of Weak Ties: A Network Theory Revisited," Sociological Theory, **1**(1983), p. 201.

[102] Flache, A., and Macy, M. W., 1996, "The weakness of strong ties: Collective action failure in a highly cohesive group," The Journal of Mathematical Sociology, **21**(1-2), pp. 3–28.

[103] Flache, A., 2002, "The rational weakness of strong ties: Failure of group solidarity in a highly cohesive group of rational agents," The Journal of Mathematical Sociology, **26**(3), pp. 189–216.

[104] Dong, A., Kleinsmann, M. S., and Deken, F., 2013, "Investigating design cognition in the construction and enactment of team mental models," Design Studies, **34**(1), pp. 1–33.

[105] Austin-Breneman, J., Yu, B. Y., and Yang, M. C., 2014, "Biased Information Passing Between Subsystems Over Time in Complex System Design," Volume 7: 2nd Biennial International Conference on Dynamics for Design; 26th International Conference on Design Theory and Methodology, ASME, Buffalo, NY, p. V007T07A023.

[106] Gooding, R. Z., and Wagner, J. A., 1985, "A Meta-Analytic Review of the Relationship between Size and Performance: The Productivity and Efficiency of Organizations and Their Subunits," Administrative Science Quarterly, **30**(4), pp. 462–481.

[107] Markham, S. E., Danserau, F., and ALutto, J. A., 1982, "Group Size and Absenteeism Rates: A Longitudinal Analysis," The Academy of Management Journal, **25**(4), pp. 921–927.

[108] Mullen, B., Symons, C., Hu, L.-T., and Salas, E., 1989, "Group Size, Leadership Behavior, and Subordinate Satisfaction," The Journal of General Psychology, **116**(May), pp. 155–170.

[109] Amason, A. C., and Sapienza, H. J., 1997, "The Effects of Top Management Team Size and interaction Norms on Cognitive and Affective Conflict," Journal of Management, **23**(4), pp. 495–516.

[110] Magjuka, R. J., and Baldwin, T. T., 2006, "Team-Based Employee Involvement Programs: Effects of Design and Administration," Personnel Psychology, **44**(4), pp. 793–812.

[111] Yetton, P. W., and Bottger, P. C., 1982, "Individual versus group problem solving: An empirical test of a best-member strategy," Organizational Behavior and Human Performance, **29**(3), pp. 307–321.

[112] Stewart, G. L., 2006, "A Meta-Analytic Review of Relationships Between Team Design Features and Team Performance," Journal of Management, **32**(1), pp. 29–55.

[113] Kozlowski, S. W. J., and Klein, K. J., 2000, "A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes," Multilevel theory, research and methods in organizations: Foundations, extensions, and new directions, S.W.J. Kozlowski, and K.J. Klein, eds., Jossey-Bass, San Francisco, CA, pp. 3–90.

[114] Qurashi, M. M., 1993, "Dependence of Publication-Rate on Size of Some University Groups and Departments in U.K. and Greece in Comparison With N.C.I., USA," Scientometrics, **27**(1), pp. 19–38.

[115] Brooks, F., 1995, The Mythical Man-Month, Addison-Wesley Longman Publishing, Boston, MA.

[116] Di Penta, M., Harman, M., Antoniol, G., and Qureshi, F., 2007, "The effect of communication overhead on software maintenance project staffing: A search-based approach," 2007 IEEE International Conference on Software Maintenance, pp. 315–324.

[117] Blackburn, J. D., Lapre, M. A., and Van Wassenhove, L. N., 2006, "Brooks' Law Revisited: Improving Software Productivity by Managing Complexity," SSRN Electronic Journal.

[118] McComb, C., Goucher-Lambert, K., and Cagan, J., 2015, "Fairness and Manipulation: An Empirical Study of Arrow's Impossibility Theorem," International Conference on

Engineering Design, Milan, Italy, pp. 267–276.

[119] He, J., Butler, B., and King, W., 2007, "Team Cognition: Development and Evolution in Software Project Teams," Journal of Management Information Systems, **24**(2), pp. 261–292.

[120] Pinto, M., 1990, "Project team communication and cross-functional cooperation in new program development," Journal of Product Innovation Management, **7**(3), pp. 200–212.

[121] Smith, K. G., Smith, K. A., Olian, J. D., Sims, H. P., O'Bannon, D. P., and Scully, J. A., 1994, "Top Management Team Demography and Process: The Role of Social Integration and Communication," Administrative Science Quarterly, **39**(3), p. 412.

[122] Espinosa, J. A., Nan, N., and Carmel, E., 2015, "Temporal Distance, Communication Patterns, and Task Performance in Teams," Journal of Management Information Systems, **32**(1), pp. 151–191.

[123] Patrashkova-Volzdoska, R. R., McComb, S. A., Green, S. G., and Compton, W. D., 2003, "Examining a curvilinear relationship between communication frequency and team performance in cross-functional project teams," IEEE Transactions on Engineering Management, **50**(3), pp. 262–269.

[124] Patrashkova, R. R., and McComb, S. A., 2004, "Exploring why more communication is not better: insights from a computational model of cross-functional teams," Journal of Engineering and Technology Management, **21**(1-2), pp. 83–114.

[125] Dinar, M., Park, Y.-S., Shah, J. J., and Langley, P., 2015, "Patterns of Creative Design: Predicting Ideation From Problem Formulation," Volume 7: 27th International Conference on Design Theory and Methodology, ASME, p. V007T06A006.

[126] Dinar, M., Shah, J., Hunt, G., Campana, E., and Langley, P., 2011, "Towards a Formal Representation Model of Problem Formulation in Design," Volume 9: 23rd International Conference on Design Theory and Methodology; 16th Design for Manufacturing and the Life Cycle Conference, ASME, pp. 263–272.

[127] Shah, J. J., Smith, S. M., and Vargas-Hernandez, N., 2003, "Metrics for measuring ideation effectiveness," Design Studies, **24**(2), pp. 111–134.

[128] Sen, P., and Yang, J.-B., 2012, Multiple Criteria Decision Support in Engineering Design, Springer Sciences and Business Media.

[129] Myers, J. L., Well, A., and Lorch, R. F., 2010, Research design and statistical analysis, Routledge.

[130] Havlin, S., Buldyrev, S. V, Goldberger, A. L., Mantegna, R. N., Ossadnik, S. M., Peng, C. K., Simons, M., and Stanley, H. E., 1995, "Fractals in Biology and Medicine," Chaos,

Solitons & Fractals, **6**(94), pp. 171–201.

[131] Shanker, O., 2006, "Random matrices, generalized zeta functions and self-similarity of zero distributions," Journal of Physics A: Mathematical and General, **39**(45), pp. 13983–13997.

[132] Cagan, J., Shimada, K., and Yin, S., 2002, "A survey of computational approaches to three-dimensional layout problems," Computer-Aided Design, **34**(8), pp. 597–611.

[133] Sorkin, G., 1991, "Efficient Simulated Annealing on Fractal Energy Landscapes," Algorithmica, **6**, pp. 367–418.

[134] Gneiting, T., and Schlather, M., 2001, "Stochastic models which separate fractal dimension and Hurst effect," SIAM Review, **46**(2), pp. 269–282.

[135] Hurst, H. E., 1957, "A Suggested Statistical Model of some Time Series which occur in Nature," Nature, **180**(4584), pp. 494–494.

[136] Izenman, A. J., 1975, "Reduced-rank regression for the multivariate linear model," Journal of Multivariate Analysis, **5**(2), pp. 248–264.

[137] Yumer, M. E., Chaudhuri, S., Hodgins, J. K., and Kara, L. B., 2015, "Semantic shape editing using deformation handles," ACM Transactions on Graphics, **34**(4), pp. 86:1–86:12.

[138] Li, X., and Schmidt, L., 2004, "Grammar-Based Designer Assistance Tool for Epicyclic Gear Trains," Journal of Mechanical Design, **126**(5), p. 895.

[139] Starling, A. C., and Shea, K., 2005, "A Parallel Grammar for Simulation-Driven Mechanical Design Synthesis," Volume 2: 31st Design Automation Conference, Parts A and B, ASME, pp. 427–436.

[140] Swantner, A., and Campbell, M. I., 2012, "Topological and parametric optimization of gear trains," Engineering Optimization, **44**(11), pp. 1351–1368.

[141] Vempati, C., and Campbell, M. I., 2007, "A Graph Grammar Approach to Generate Neural Network Topologies," ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, pp. DETC2007–34588.

[142] Kurtoglu, T., and Campbell, M. I., 2006, "A Graph Grammar Based Framework for Automated Concept Generation," International Design Conference, pp. 61–68.

[143] Königseder, C., Shea, K., and Campbell, M. I., 2013, "Comparing a Graph-Grammar Approach to Genetic Algorithms for Computational Synthesis of PV Arrays," CIRP Design 2012, pp. 193–203.

[144] Kiff, R., and Campbell, M., 2015, “An Agent-Based Approach to Synthesizing Structures,” Volume 7: 27th International Conference on Design Theory and Methodology, ASME, p. V007T06A015.

[145] Königseder, C., and Shea, K., 2014, “Systematic Rule Analysis of Generative Design Grammars,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **28**(3), pp. 227–238.

[146] Dennis, A. R., and Gallupe, B. R., 1992, “A History of Group Support Systems Empirical Research: Lessons Learned and Future Directions,” Group Spport Systems: New Perspectives.

[147] Valacich, J. S., Dennis, A. R., and Connolly, T., 1994, “Idea Generation in Computer-Based Groups: A New Ending to an Old Story,” Organizational Behavior and Human Decision Processes, **57**(3), pp. 448–467.

[148] Hinsz, V. B., Tindale, R. S., and Vollrath, D. a, 1997, “The emerging conceptualization of groups as information processors.,” Psychological bulletin, **121**(1), pp. 43–64.

[149] Salomon, G., and Globerson, T., 1989, “When teams do not function the way they ought to,” International Journal of Educational Research, **13**(1), pp. 89–99.

[150] Diehl, M., and Stroebe, W., 1991, “Productivity loss in idea-generating groups: Tracking down the blocking effect.,” Journal of Personality and Social Psychology, **61**(3), pp. 392–403.

[151] Nijstad, B. a, and Stroebe, W., 2006, “How the group affects the mind: a cognitive model of idea generation in groups,” Personality and social psychology review : an official journal of the Society for Personality and Social Psychology, Inc, **10**(3), pp. 186–213.

[152] Mantei, M., 1981, “The effect of programming team structures on programming tasks,” Communications of the ACM, **24**(3), pp. 106–113.

[153] Olguín Olguín, D., Waber, B. N., Kim, T., Mohan, A., Ara, K., and Pentland, A., 2009, “Sensible organizations: Technology and methodology for automatically measuring organizational behavior,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **39**(1), pp. 43–55.

[154] Luong, A., and Rogelberg, S. G., 2005, “Meetings and More Meetings: The Relationship Between Meeting Load and the Daily Well-Being of Employees.,” Group Dynamics: Theory, Research, and Practice, **9**(1), pp. 58–67.

[155] Zijlstra, F. R. H., Roe, R. A., Leonora, A. B., and Krediet, I., 1999, “Temporal factors in mental work: Effects of interrupted activities,” Journal of Occupational and Organizational Psychology, **72**, pp. 163–185.

[156] Zohar, D., 1999, "When things go wrong: The effect of daily work hassles on effort, exertion and negative mood," Journal of Occupational & Organizational Psychology, **72**, pp. 265–283.

[157] Wright, D. B., 2007, "Calculating nominal group statistics in collaboration studies," Behavior research methods, **39**(3), pp. 460–70.

[158] Keene, O. N., 1995, "The log transformation is special.," Statistics in medicine, **14**(8), pp. 811–819.

[159] Zajonc, R. B., 1965, "Social Facilitation," Science, **149**, pp. 269–274.

[160] Zajonc, R. B., and Sales, S. M., 1966, "Social facilitation of dominant and subordinate responses," Journal of Experimental Social Psychology, **2**(2), pp. 160–168.

[161] Becchetti, L., Pelloni, A., and Rossetti, F., 2008, "Relational goods, sociability, and happiness," Kyklos, **61**(3), pp. 343–363.

[162] Mohammed, S., 2001, "Toward an understanding of cognitive consensus in a group decision-making context," The Journal of Applied Behavioral Science, **37**(4), pp. 408–425.

[163] Mohammed, S., and Ringseis, E., 2001, "Cognitive diversity and consensus in group decision making: The role of inputs, processes, and outcomes," Organizational behavior and human decision processes, **85**(2), pp. 310–335.

[164] Weinstein, N. D., and Klein, W. M., 1996, "Unrealistic Optimism: Present and Future," Journal of Social and Clinical Psychology, **15**(1), pp. 1–8.

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

[166] Cagan, J., Dinar, M., Shah, J. J., Leifer, L., Linsey, J., Smith, S., and Vargas-Hernandez, N., 2013, "Empirical Studies of Design Thinking: Past, Present, Future," Volume 5: 25th International Conference on Design Theory and Methodology; ASME 2013 Power Transmission and Gearing Conference, ASME, p. V005T06A020.

[167] Tseng, I., Cagan, J., and Kotovsky, K., 2012, "Concurrent Optimization of Computationally Learned Stylistic Form and Functional Goals," Journal of Mechanical Design, **134**(11), pp. 111006–1 – 111006–11.

[168] Goel, A. K., Vattam, S., Wiltgen, B., and Helms, M., 2012, "Cognitive, collaborative, conceptual and creative — Four characteristics of the next generation of knowledge-based CAD systems: A study in biologically inspired design," Computer-Aided Design, **44**(10), pp. 879–900.

[169] Egan, P., Cagan, J., Schunn, C., and LeDuc, P. R., 2015, "Synergistic human-agent methods for deriving effective search strategies: the case of nanoscale design," Research in Engineering Design, **26**(2), pp. 145–169.

[170] Szykman, S., and Cagan, J., 1995, "A Simulated Annealing-Based Approach to Three-Dimensional Component Packing," Journal of ….

[171] Königseder, C., and Shea, K., 2015, "A Method for Visualizing the Relations Between Grammar Rules, Performance Objectives and Search Space Exploration in Grammar-Based Computational Design Synthesis," Volume 2A: 41st Design Automation Conference, ASME, p. V02AT03A007.

[172] Hiroyasu, T., Miki, M., Ogura, S., Aoi, K., Yoshida, T., Okamoto, Y., and Dongarra, J., 2002, "Energy Minimization of Protein Tertiary Structure by Parallel Simulated Annealing using Genetic Crossover SA and GA Parallel SA using Genetic," 2002 Genetic and Evolutionary Computation Conference (GECCO 2002) Workshop Program, pp. 49–51.

[173] Aydin, M., and Fogarty, T., 2004, "A simulated annealing algorithm for multi-agent systems: a job-shop scheduling application," Journal of Intelligent Manufacturing.

[174] Bilbao, M., and Alba, E., 2009, "Simulated annealing for optimization of wind farm annual profit," 2009 2nd International Symposium on Logistics and Industrial Informatics, LINDI 2009, **0**(2), pp. 1–5.

[175] Ingber, L., 1996, "Adaptive simulated annealing (ASA): lessons learned," Controls and Cybernetics, **25**(1), pp. 33–54.

[176] Granville, V., Krivanek, M., and Rasson, J.-P., 1994, "Simulated Annealing: A Proof of Convergence," IEEE Transactions on pattern Analysis and Machine Intelligence, **16**(6), pp. 652–656.

[177] Azizi, N., and Zolfaghari, S., 2004, "Adaptive temperature control for simulated annealing: a comparative study," Computers & Operations Research, **31**(14), pp. 2439–2451.

[178] Huang, M. D., Romeo, F., and Sangiovanni-Vincentelli, A. L., 1986, "An efficient general cooling schedule for simulated annealing," Proceedings of the IEEE International Conference on Computer-Aided Design, pp. 381–384.

[179] Ohlidal, M., and Schwarz, J., 2004, "Hybrid parallel simulated annealing using genetic operations," 10th International Conference on Soft Computing, pp. 89–94.

[180] Zhong, Y., Ning, J., and Zhang, H., 2012, "Multi-agent simulated annealing algorithm based on particle swarm optimisation algorithm," International Journal of Computer

Applications in Technology, **43**(4), pp. 335–342.

[181] Zhong, Y., Wang, L., Wang, C., and Zhang, H., 2012, "Multi-agent simulated annealing algorithm based on differential evolution algorithm," International Journal of Bio-Inspired Computation, **4**(4), pp. 217–228.

[182] Landry, L. H., and Cagan, J., 2011, "Protocol-Based Multi-Agent Systems: Examining the Effect of Diversity, Dynamism, and Cooperation in Heuristic Optimization Approaches," Journal of Mechanical Design, **133**(2), p. 021001.

[183] Alba, E., and Troya, J. M., 1999, "A survey of parallel distributed genetic algorithms," Complexity, **4**, pp. 31–52.

[184] Wales, D., and Doye, J. P. K., 1997, "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms," Journal of Physical Chemistry A, **101**, pp. 5111–5116.

[185] Eberhart, R. C., and Shi, Y., 2001, "Particle swarm optimization: developments, applications and resources," Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), IEEE, pp. 81–86.

[186] Jones, D. R., Schonlau, M., and William, J., 1998, "Efficient Global Optimization of Expensive Black-Box Functions," Journal of Global Optimization, **13**, pp. 455–492.

[187] Kearfott, R. B., and Kreinovich, V., eds., 1996, Applications of Interval Computations, Springer US, Boston, MA.

[188] Yin, S., and Cagan, J., 2000, "An Extended Pattern Search Algorithm for 3D Component Layout," Journal of Mechanical Design, **122**(1), pp. 102–108.

[189] Papalambros, P. Y., and Wilde, D. J., 2000, Principles of optimal design: modeling and computation, Cambridge University Press.

[190] The Mathworks Inc., 2014, "MATLAB."

[191] The Mathworks Inc., 2014, "Global Optimization Toolbox."

[192] Bäck, T., 1996, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press.

[193] Griewank, A. O., 1981, "Generalized Descent for Global Optimization," Journal of Optimization Theory and Applications, **34**(1), pp. 11–39.

[194] Mühlenbein, H., Schomisch, M., and Born, J., 1991, "The parallel genetic algorithm as function optimizer," Parallel Computing, **17**, pp. 619–632.

[195] Clegg, B. A., Digirolamo, G. J., and Keele, S. W., 1998, "Sequence learning," Trends in

Cognitive Sciences, **2**(8), pp. 275–281.

[196] Chase, W. G., and Simon, H. A., 1973, “Perception in Chess,” Cognitive Psychology, **4**(1), pp. 55–81.

[197] Egan, D. E., and Schwartz, B. J., 1979, “Chunking in recall of symbolic drawings,” Memory & Cognition, **7**(2), pp. 149–58.

[198] Reitman, J. S., 1976, “Skilled perception in Go: Deducing memory structures from inter-response times,” Cognitive Psychology, **8**(3), pp. 336–356.

[199] Moss, J., Cagan, J., and Kotovsky, K., 2004, “Learning from design experience in an agent-based design system,” Research in Engineering Design, **15**, pp. 77–92.

[200] Simon, H. A., and Kotovsky, K., 1963, “Human acquisition of concepts for sequential patterns,” Psychological review, **70**(6), pp. 534–546.

[201] Kotovsky, K., and Simon, H. A., 1973, “Empirical tests of a theory of human acquisition of concepts for sequential patterns,” Cognitive Psychology, **4**(3), pp. 399–424.

[202] Novick, L. R., and Tversky, B., 1987, “Cognitive constraints on ordering operations: the case of geometric analogies.,” Journal of experimental psychology. General, **116**(1), pp. 50–67.

[203] Curran, T., and Keele, S. W., 1993, “Attentional and Nonattentional Forms of Sequence Learning,” Journal of Experimental Psychology: Learning, Memory, and Cognition, **19**, pp. 189–202.

[204] Atman, C. J., Adams, R. S., Cardella, M. E., Turns, J., Mosborg, S., and Saleem, J., 2007, “Engineering Design Processes: A Comparison of Students and Expert Practitioners,” Journal of Engineering Education, **96**(4), pp. 359–379.

[205] Goldschmidt, G., and Rodgers, P. A., 2013, “The design thinking approaches of three different groups of designers based on self-reports,” Design Studies, **34**(4), pp. 454–471.

[206] Radcliffe, D. F., and Lee, T. Y., 1989, “Design methods used by undergraduate engineering students,” Design Studies, **10**(4), pp. 199–207.

[207] Todd, D., 1997, “Multiple Criteria Genetic Algorithms in Engineering Design and Operation,” University of Newcastle.

[208] Rogers, J., 1996, “DeMAID/GA - An enhanced design manager’s aid for intelligent decomposition,” 6th Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics.

[209] Sen, C., Ameri, F., and Summers, J. D., 2010, “An Entropic Method for Sequencing Discrete Design Decisions,” Journal of Mechanical Design, **132**(10), p. 101004.

[210] Waldron, M. B., and Waldron, K. J., 1988, “A time sequence study of a complex mechanical system design,” Design Studies, **9**(2), pp. 95–106.

[211] Meier, C., Yassine, A. A., and Browning, T. R., 2007, “Design Process Sequencing With Competent Genetic Algorithms,” Journal of Mechanical Design, **129**(6), pp. 566–585.

[212] Gero, J. S., 1990, “Design Prototypes : A Knowledge Representation Schema for Design,” AI Magazine, **11**(4), pp. 26–36.

[213] Bhatta, S. R., and Goel, A. K., 1994, “Discovery of physical principles from design experiences,” International Journal AI EDAM (AI for Engineering Design, Analysis, and Manufacturing), (July 1992), pp. 1–22.

[214] Neill, T. M., Gero, J. S., and Warren, J., 1998, “Understanding conceptual electronic design using protocol analysis,” Research in Engineering Design, **10**(3), pp. 129–140.

[215] Kan, J. W. T., and Gero, J. S., 2011, “Comparing Designing Across Different Domains : an Exploratory Case Study,” Design, (August).

[216] Yu, R., and Gero, J. S., 2015, “An empirical foundation for design patterns in parametric design,” Proceedings of the 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA 2015, (April), pp. 1–9.

[217] Gero, J. S., Kan, J. W., and Pourmohamadi, M., 2011, “Analysing Design Protocols : Development of Methods and Tools,” Research into Design, pp. 3–10.

[218] Gero, J. S., and Peng, W., 2009, “Understanding behaviors of a constructive memory agent: A markov chain analysis,” Knowledge-Based Systems, **22**(8), pp. 610–621.

[219] Vale, C. A. W., and Shea, K., 2003, “A Machine Learning-Based Approach to Accelerating Computational Design Synthesis,” Proceedings of the 8th International Conference On Engineering Design, pp. 445–446.

[220] Stroock, D. W., 2005, An Introduction to Markov Processes, Springer Science and Business Media.

[221] Seneta, E., 2006, “Markov and the creation of Markov chains,” Markov Anniversary Meeting 2006, pp. 1–20.

[222] Scherr, A. L., 1962, “An Analysis of Time-Shared Computer Systems,” Massachusetts Institute of Technology.

[223] Page, L., Brin, S., Motwani, R., and Winograd, T., 1999, The PageRank Citation Ranking: Bringing Order to the Web., Stanford InfoLab.

[224] Tamir, A., 1998, Applications of Markov Chains in Chemical Engineering, Elsevier.

[225] Gero, J. S., and Kan, J. W., 2009, “Learning to collaborate during team designing: some preliminary results from measurement-based tools,” ICORD 09: Proceedings of the 3rd International Conference on Research into Design Engineering, Bangalore, India, pp. 560–567.

[226] Gero, J. S., and Kan, J. W., 2011, “Learning to collaborate during team designing: quantitative measurement,” ICORD 11: Proceedings of the 3rd International Conference on Research into Design Engineering, pp. 978–981.

[227] Raftery, A. E., 1985, “A Model for High-Order Markov Chains,” Journal of the Royal Statistical Society, Series B (Methodological), **47**(3), pp. 528–539.

[228] Baum, L. E., and Petrie, T., 1966, “Statistical Inference for Probabilistic Functions of Finite State Markov Chains,” The Annals of Mathematical Statistics, **37**(6), pp. 1554–1563.

[229] Baum, L. E., and Eagon, J. A., 1967, “An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology,” Bulletin of the American Mathematical Society, **73**(3), pp. 360–364.

[230] Baum, L. E., and Sell, G., 1968, “Growth transformations for functions on manifolds,” Pacific Journal of Mathematics, **27**(2), pp. 211–227.

[231] Baum, L. E., Petrie, T., Soules, G., and Weiss, N., 1970, “A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains,” The Annals of Mathematical Statistics, **41**(1), pp. 164–171.

[232] Rabiner, L. R., 1989, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of the IEEE, **77**(2), pp. 257–286.

[233] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D., 1994, “Hidden Markov models in computational biology. Applications to protein modeling.,” Journal of molecular biology, **235**(5), pp. 1501–1531.

[234] Gonzalez, A. M., Roque, A. M. S., and Garcia-Gonzalez, J., 2005, “Modeling and Forecasting Electricity Prices with Input/Output Hidden Markov Models,” IEEE Transactions on Power Systems, **20**(1), pp. 13–24.

[235] White, B., Blaylock, N., and Ladislau, B., “Analyzing Team Actions with Cascading HMM Team action analysis : Bounding Overwatch.”

[236] Arlot, S., and Celisse, A., 2010, “A survey of cross-validation procedures for model selection,” **4**, pp. 40–79.

[237] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., 1998, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.

[238] Cagan, J., and Vogel, C. M., 2012, Creating breakthrough products: Innovation from product planning to program approval, FT Press.

[239] Weingart, L. R., Cronin, M. A., Houser, C. J. S., Cagan, J., and Vogel, C. M., 2005, "Functional diversity and conflict in cross-functional product development teams: Considering representational gaps and task characteristics," Understanding Teams, L.L. Neider, and C.A. Schriesheim, eds., Information Age Publishing, Greenwich, CT, pp. 89–110.

[240] Luo, X., Slotegraaf, R. J., and Pan, X., 2006, "Cross-Functional 'Coopetition': The Simultaneous Role of Cooperation and Competition Within Firms," Journal of Marketing, **70**(2), pp. 67–80.

[241] Wall, S., and Lepsinger, R., 1994, "Cross-Functional Obstacles," Training, **5**(2), pp. 125–126.

[242] Tabrizi, B., 2015, "75% of Cross-Functional Teams Are Dysfunctional," Harvard Business Review.

[243] Austin-Breneman, J., Yu, B. Y., and Yang, M. C., 2015, "Biased Information Passing Between Subsystems Over Time in Complex System Design," Journal of Mechanical Design, **138**(1), p. 011101.

[244] Scott, M. J. M., and Antonsson, E. K. E., 1996, "Formalisms for negotiation in engineering design," 8th International Conference on Design Theory and Methodology ASME, pp. 165–182.

[245] Eiben, A., and Smit, S., 2012, Evolutionary Algorithm Parameters and Methods to Tune Them, Springer Berlin Heidelberg, Berlin, Heidelberg.

[246] Bishop, C. M., and Lasserre, J., 2007, "Generative or discriminative? getting the best of both worlds," Bayesian Statistics, J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, and M. West, eds., pp. 3–24.