Mining Process Heuristics From Designer Action Data Via Hidden Markov Models

Christopher McComb
Mem. ASME
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15232
e-mail: ccm@cmu.edu

Jonathan Cagan
Fellow ASME
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15232
e-mail: cagan@cmu.edu

Kenneth Kotovsky
Department of Psychology,
Carnegie Mellon University,
Pittsburgh, PA 15232
e-mail: kotovsky@cmu.edu

Configuration design problems, characterized by the assembly of components into a final desired solution, are common in engineering design. Various theoretical approaches have been offered for solving configuration type problems, but few studies have examined the approach that humans naturally use to solve such problems. This work applies data-mining techniques to quantitatively study the processes that designers use to solve configuration design problems. The guiding goal is to extract beneficial design process heuristics that are generalizable to the entire class of problems. The extraction of these human problem-solving heuristics is automated through the application of hidden Markov models to the data from two behavioral studies. Results show that designers proceed through four procedural states in solving configuration design problems, roughly transitioning from topology design to shape and parameter design. High-performing designers are distinguished by their opportunistic tuning of parameters early in the process, enabling a more effective and nuanced search for solutions. [DOI: 10.1115/1.4037308]

1 Introduction

The selection and assembly of specific components to accomplish a well-defined objective is a familiar task in engineering, commonly referred to as configuration design [1]. Although a variety of theoretical approaches to solving configuration design problems have been developed, little research has been conducted to observe how humans naturally approach such problems. Thus, the overarching goal of this work is to examine the results of two behavioral studies of configuration design [2,3] to extract beneficial heuristics. The approach used to extract these heuristics is formal, novel, and data-driven. Specifically, we mine the behavioral study data using hidden Markov models to extract well-defined descriptions of design behavior and process. It is shown that hidden Markov models trained on the data are robust enough to indicate strong similarities between the two behavioral studies, and are also sensitive enough to discover and clarify the procedural differences between high- and low-performing designers. A comparison of these performance-differentiated process models provides insights leading to effective and general configuration design heuristics. The extraction of these guiding heuristics is necessary because many configuration design problems do not admit well-defined objective functions. This precludes their solution through numerical algorithms and necessitates direct processing by designers who may benefit from the heuristics found in this work.

This work specifically mines the data generated by humans while designing either a truss structure or an internet-connected cooling system. For both problems, the solving approach used by participants generally conformed to a propose–critique–modify methodology, one frequently utilized for configuration design [4]. This method begins with an initial solution (propose), evaluates the solution against constraints and objectives (critique), and changes the solution to reduce constraint violations or improve objectives (modify). Propose–critique–modify and other methods [5] provide a structured approach for searching the solution space associated with a given configuration problem. However, the solution space of configuration design problems branches polynomially [1], meaning that it tends to be both complex and large. Methods like propose–critique–modify cannot search efficiently unless they are guided by insights about the problem or heuristics that reduce the effective size of the solution space. This paper leverages a data-mining approach to accomplish the extraction and codification of beneficial heuristics used by human designers.

Other work has examined the patterns of rule-based operations that are used by humans to solve a truss-type configuration problem. The rule-based operations for truss design problems are typically broken into three classes: topology operations (which modify the connectedness of the truss), spatial operations (which change the location of existing joints), and parameter operations (which modify the characteristics of structural members). Behavioral study participants were shown to predominantly use topology operations during the early phases of design, and progressively use more spatial and parameter operations in later phases [6]. A comparison of high- and low-performing teams in that study showed that high-performing teams used a smaller proportion of topology operations throughout the study [6]. However, both high- and low-performing teams progressively introduced more parameter and spatial operations at approximately the same rate. Statistical models were used to examine the sequential patterns employed during configuration design [7]. It was revealed that strong sequencing of operations was important for achieving high quality designs. Specifically, topology operations were often applied together in specific sequences with other topology operations, while spatial and parameter operations were applied separately. It has also been noted that strong spatial cognition abilities are correlated to better outcomes in configuration design tasks [8]. Studies that examine configuration design can be challenging for participants, requiring them to not only create models but also interact with and modify those models to meet specific objectives.

Designers are known to frequently switch between embodiment and detail design activities in the later stages of the design process [9]. This type of switching between different levels of detail or abstraction can be an indication of opportunistic design behavior [10], which is characterized by responsiveness to emergent requirements or the characteristics of partial solutions [11]. Bender and Blessing [12] linked the concept of opportunism in design to the use of prescriptive methodologies in early

Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received February 20, 2017; final manuscript received June 5, 2017; published online October 2, 2017. Assoc. Editor: Harrison M. Kim.

conceptual design. Specifically, it was demonstrated that a hierarchical, phase-oriented approach (i.e., a highly structured prescriptive methodology) produced final design solutions with worse quality than those produced with a completely unstructured trial-and-error approach. The best solutions resulted from participants who used a balanced approach that merged opportunistic solving with a small degree of hierarchical structure [12]. Configuration design can involve different degrees of abstraction or detail, ranging from the connectivity of components (more abstract) to the tuning of component parameters (more detailed). Thus, opportunistic switching between these levels may play an important role in the data mined in this work.

Little research has been conducted to observe how humans naturally approach configuration problems. However, such descriptive research is a necessary initial step before prescriptive heuristics can be codified. In this work, the treatment of designer activity as a hidden Markov process makes it possible to infer data-driven process models directly from a log of designer activity, providing a descriptive basis. The resultant hidden Markov models can then be used to compare the design process used on different configuration design problems, as well as comparing the process used by high- and low-performing designers. This comparison aims at extracting prescriptive recommendations, with the ultimate goal of directing designers toward configuration design procedures that reduce the effective size of the search space. This paper is organized as follows: Section 2 provides background on hidden Markov models and their uses. Section 3 reviews the two human behavioral studies that are analyzed in the remainder of the paper. Section 4 explains the analytical methods used to analyze the data from the two human studies. Section 5 details the results of applying hidden Markov methodology to configuration design data, Sec. 6 provides a comparative discussion of the resulting models from the two behavioral studies, and Sec. 7 summarizes conclusions and identifies directions for future work. Ultimately, this work demonstrates that the patterns used by designers during configuration design may be problem-independent, and that there may exist heuristics that improve performance within the general class of configuration problems.

2 Hidden Markov Model

A hidden Markov model describes a stochastic process in which a system transitions between a finite number of discrete states which cannot be directly observed [13]. Instead, the states probabilistically output tokens that can be observed. Each state is assumed to have a probability distribution over the set of possible output tokens, and one token is emitted from the system at every timestep. Therefore, the sequence of tokens that is produced by a hidden Markov model gives information about the structure of the hidden states.

The parameters that define the hidden Markov model are the transition matrix, $\boldsymbol{T}$, and the emission matrix, $\boldsymbol{E}$. The transition matrix contains the probability of transitioning to a future state from a current state, where the value of T_{ij} is the probability of transitioning from state i to state j. The transition matrix has size $k \times k$, where k is the number of hidden states. The emission matrix contains the probability that a token will be emitted from a given state, where E_{ij} is the probability that state i will emit token j. The emission matrix has size $k \times m$, where m is the number of tokens, and k is the number of hidden states.

The mathematics describing hidden Markov models were established by Baum et al. [17]. One of the first practical uses of hidden Markov models was for speech recognition [18], but they have also been utilized in fields as diverse as protein modeling [19], economic forecasting [20], team military tactics [21], and cognitive skill acquisition [22]. Figure 1 shows a hidden Markov model with three hidden states and three emission tokens. Hidden states are shown by circles, and emission tokens are shown with squares. The transitions between hidden states are shown with solid

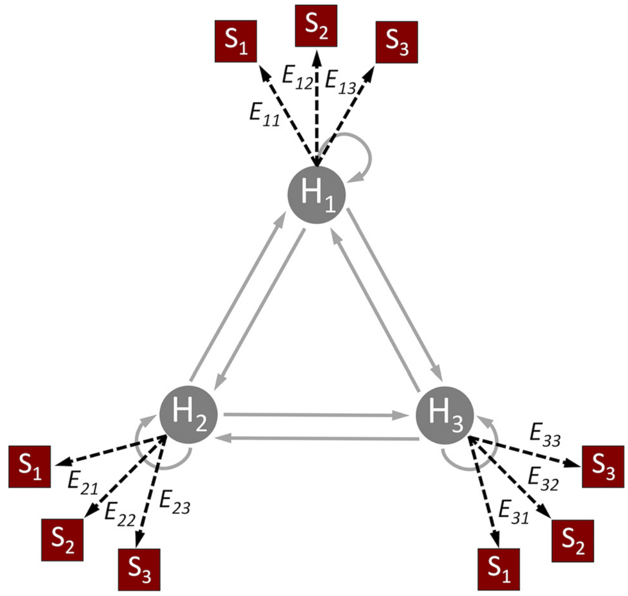

Fig. 1 Example of a hidden Markov model with three states and three emission tokens

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.

In the application explored in this work, the tokens emitted by the model are design operations. By treating the design operations probabilistically in this way, the hidden states that constitute the model represent the underlying cognitive or procedural states that the designer transitions through during the design process. Information on design quality is used here to sort behavioral data prior to training hidden Markov models, enabling the creation of models that specifically correspond to high performance. However, our approach intentionally refrains from incorporating quality data in the models themselves. Design problems often do not have well-defined objective functions, and designers may spend extended periods of time working on incomplete solutions that cannot yet be evaluated. Avoiding the use of solution quality information in the models developed here ensures that the extracted heuristics can be utilized by designers who are facing ill-defined problems or are working on incomplete solutions. Further, it will be shown that the models created here encode procedural differences that may be useful for predicting the performance of designers.

Hidden Markov models have been used previously to analyze the cognitive procedural stages in problem-solving tasks [22,23]. Since design is often viewed as problem-solving, the use of hidden Markov models in the current application is justified. An important feature of the previous work is that the model transition probabilities were constrained in such a way that the model could only proceed linearly through the hidden states. However, the procedural aspects of design are often nonlinear, and designers often return to earlier design stages and iterate on solutions [24–26]. Therefore, the current application requires a more general modeling style that enables these nonlinear aspects. To that end, we employ hidden Markov models with no directional constraints on state transitions.

A key assumption in the use of this type of model is that the state of a system depends only on the immediately previous state (rather than multiple previous states). This is known as first-order sequencing. While this may initially seem to be a limitation, other work has demonstrated that including multiple previous states does not significantly increase model veracity for Markov-based models in design applications [7]. In addition, evidence from

Table 1 Summary of key characteristics from behavioral studies

	Truss study	Cooling system study
Problem domain	Structural	Thermal/fluid
No. of operation types	7	9
Participant discipline	Mechanical engineering	Mechanical engineering
Participant academic level	Senior undergraduate	Senior undergraduate and graduate
Total participants	48	67
Operations per participant	430.1 ± 123.5	50 ± 0 (fixed number)

studies of humans solving tavern puzzles has shown that first-order sequencing is important for effective problem-solving [27].

3 Data Sets

The data sets analyzed in this work were derived from two previously conducted behavioral studies. The first study tasked engineering students with the design of a truss structure [2], and the second study tasked a different group of engineering students with the design of an internet-connected home cooling system [3]. A review of both studies is given in this section, with an overview of key characteristics provided in Table 1.

Both behavioral studies were conducted in teams, but the current work treats data from these studies at the individual level. This type of individual-level analysis can be performed on the team-based data for two reasons. First, a separate series of operations was logged independently for every participant, rather than collecting aggregated data at the team level. Second, the fraction of time spent working individually was much larger than the fraction of time spent interacting with teammates for both studies, so patterns of behavior resulted primarily from individual effort.

3.1 Data From Truss Behavioral Study. In the truss behavioral study, teams of three senior mechanical engineering students were tasked with the design of a truss structure. Restricting participation to seniors in mechanical engineering helped to ensure that all participants had basic fluency in structural mechanics concepts. The original study was also designed to test teams' responses to dynamic and changing design scenarios, so the problem statement was unexpectedly changed at two points during solving. The initial problem statement asked participants to design a truss to cross two spans and support a load at the middle of each. The first unexpected change instructed participants to consider the possibility that one of the three supports for the truss could fail due to an adversarial attack. Thus, participants were required to make their truss structurally redundant to survive such an attack. The second change introduced an area through which no members could pass, essentially requiring students to design around an obstacle. For each problem statement, participants were given a required factor-of-safety and a target mass.

Every participant was given access to a graphical truss design computer program to facilitate completion of the design task. More information on this tool can be found in Ref. [2]. Through this interface, participants could construct, analyze, and share trusses within their team. Solution quality was automatically computed and displayed to participants. The interface was also used to record a full log of the actions and operations of the participants. In constructing their trusses, participants could perform seven distinct design operations: adding a joint, removing a joint, adding a member, removing a member, moving a joint, changing the size of all members simultaneously, and changing the size of a single member. There was no limitation on the total number of joints or members that could be created, or on the number of members that could be connected to a single joint. Every participant performed an average of 400–500 such operations. A short example operation sequence is provided in Fig. 2. The truss design problem constitutes a *full configuration design* problem, per the guidelines provided by Wielinga and Schreiber [5]. This type of problem is characterized by parameterized components (e.g., size of members, location of joints), the lack of a predefined arrangement, and functional or global requirements and constraints (e.g., factor of safety, mass).

Prior to beginning their design work, each participant engaged in a 10-min computer-guided tutorial that introduced them to the functionality of the user interface and reiterated basic structural mechanics concepts. This tutorial ensured that participants had adequate knowledge of the computer interface when they started to design, removing the need to learn how to operate the interface while simultaneously creating a solution.

3.2 Data From Home Cooling System Behavioral Study. This study tasked teams of three mechanical engineering students

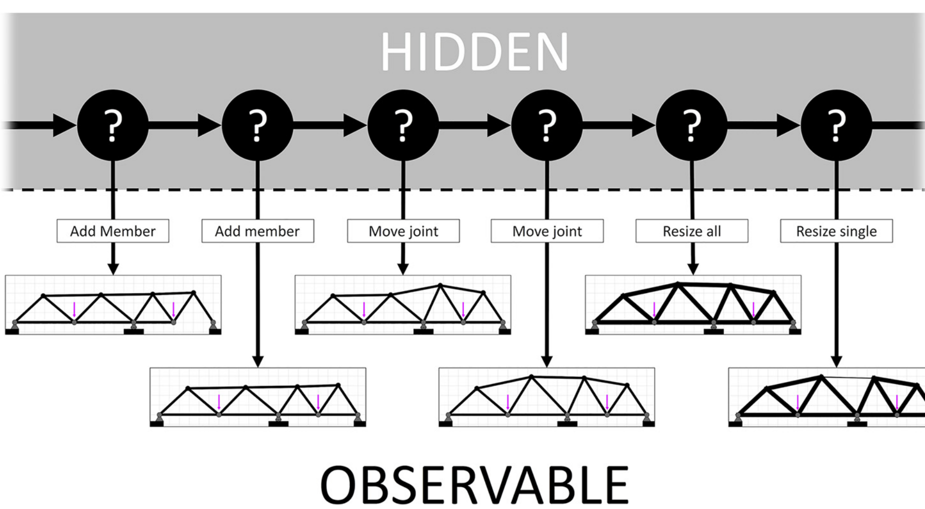

Fig. 2 Example truss operation sequences

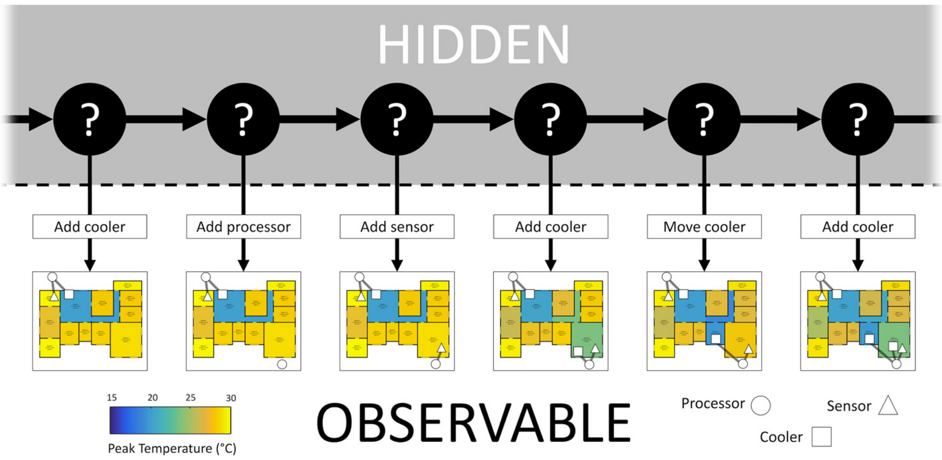

Fig. 3 Example cooling system operation sequence with shading indicating room temperatures

with the design of an internet-connected cooling system for a home. Participants were instructed to minimize the total cost of the cooling system (accounting for both the cost of the system itself and amortized operating costs) and to minimize temperature within the home. Participants were given target values for both objectives. The primary purpose of this study was to test the effect of different interaction frequencies on team performance. Three different conditions were tested with zero interaction, low interaction, and moderate interaction.

Much like the truss study, participants in this study were given access to a computer interface that allowed them to construct, analyze, and share solutions within their teams (solution quality was automatically computed and displayed). The computer interface was also used to record a full log of the actions and operations of the participants. More information about this interface can be found in Refs. [3,28].

Three fundamental product types could be used to create a cooling system: sensors, to sense the temperature in specific rooms; coolers, to reduce the temperature in specific rooms; and processors, to read the information from sensors and decide which coolers to activate and when to activate them. The connectivity of these products was limited so that any cooler or sensor could only be connected to one processor. However, a processor could be connected to any number of coolers and sensors. There were no constraints on the location at which these products could be added within the home. In constructing their systems, participants could perform nine fundamental design operations: adding a processor, adding a sensor, adding a cooler, removing a processor, removing a sensor, removing a cooler, moving a cooler, moving a sensor, and resizing a cooler. A key difference between this study and the truss behavioral study is that participants were made to perform exactly 50 design operations, thus ensuring parity between the different conditions. This is approximately an order of magnitude below the number of operations performed by participants in the truss behavioral study. Participants were made aware of this limitation at the beginning of the study, and the computer interface communicated the number of operations remaining. The cooling system design problem is also a case of *full configuration design* since components are parameterized (sensor/cooler location, cooler flow/power), the problem lacks a predefined arrangement, and global requirements and constraints are implemented (cost, temperature) [5]. A short example sequence of operations for this problem is provided in Fig. 3, with shading indicating temperature.

Participation in this study was restricted to students in mechanical engineering, which helped to ensure a common set of background knowledge. In addition, post hoc analysis of the results did not reveal a significant effect on performance based on the number of students who noted drawing upon relevant coursework during the study. Thus, it is unlikely that variations in domain-specific knowledge played a role in solution quality. In addition, before beginning to design, participants engaged in a 10-min computer-guided tutorial that introduced them to the functionality of the computer interface. This tutorial ensured that participants had adequate knowledge of the computer interface before beginning to design, mitigating the need to learn how to operate the interface while also attempting to create a viable solution.

4 Heuristic Mining Methodology

This work analyzes the data from behavioral studies using hidden Markov models. This section outlines the specific methodology used in the analysis. Section 4.1 outlines the Baum–Welch algorithm, which provides a means for computing the transmission and emission matrices for a hidden Markov model. The Baum–Welch algorithm assumes that the number of hidden states is known, but such information was not available for the current application. For that reason, Sec. 4.2 outlines the procedure used here to estimate the correct number of hidden states. Section 4.3 details the procedure used for computing high-performance and low-performance hidden Markov models.

4.1 The Baum–Welch Algorithm. If the number of hidden states (k) is known, a hidden Markov model can be trained using the Baum–Welch algorithm [17]. This algorithm uses an expectation–maximization approach [29] to provide maximum-likelihood estimates of 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). The expectation step of the Baum–Welch algorithm utilizes the forward–backward algorithm to compute the probability that every observation in the training data resulted from any state in the model [30]. The maximization step then updates the transmission and emission probabilities of the model so that the likelihood of the observed data (given the model parameters) is maximized. A more detailed account of the algorithm is given in Ref. [13].

4.2 Determining the Number of Hidden States. 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 several models with varying values of k, and then compare them in some way to select the best value. In this work, we trained several models with values of k from one to the number of operations associated with the design problem (seven for truss study, and nine for the cooling system study). 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 (design operations). Furthermore, when k is equal to the number of operations, the emission matrix becomes equal to the identity matrix, and the model collapses to a first-order Markov chain. Similarly, when $k=1$ the hidden Markov model contains only a single state, which amounts to a simple frequency distribution over operations. Both of these cases are detailed in Ref. [7].

For each value of k, models were trained using leave-one-out cross-validation [31]. For a data set consisting of n samples (each sample here consists of the data from a single participant), 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. Training a model on data from multiple individuals in this way increases the generalizability of the results. This procedure is repeated until every individual sample has been used for testing. The best value for k can be then be selected by comparing the set of trained models based on testing log-likelihood (indicative of the model's ability to represent data that it was not explicitly trained on). Specifically, we selected the lowest value of k for which the testing log-likelihood of the trained model was not significantly different from the testing log-likelihood of the model trained with maximum k. This selection criterion balances between model parsimony and descriptive accuracy by selecting the smallest number of hidden states necessary to offer a significantly accurate description of the data.

4.3 Training Models on Performance-Differentiated Data.

A key aspect of this work is the utilization of hidden Markov models to compare the procedures used by high-performing and low-performing designers. To perform this comparison, the participants from each study were segmented according to performance to produce a high-performing segment and a low-performing segment (both constituting roughly one third of the total population).

Of the 48 individuals that took part in the truss study, 15 individuals (31.25% of the population) were designated as high-performing designers and 15 individuals (31.25% of the population) were designed as low-performing designers based on an evaluation of the final and intermittent design solutions provided by their teams. A detailed account of how high- and low-performing designers were designated is available in Ref. [2]. The difference in the strength-to-weight ratio of final designs delivered by these groups was statistically significant ($F=11.72, p<0.01$).

A total of 67 individuals participated in the cooling system study. The individuals whose final design solution met or exceeded the target values for cost *and* temperature were designated as high-performing designers. The individuals whose final design solution met neither target value were designated as low-performing designers. This procedure resulted in the identification of 16 high-performing designers (23.88% of the population) and 22 low-performing designers (32.86% of the population). High-performing designers achieved solutions with significantly lower peak temperature ($F=26.01,\ p<0.001$) and significantly lower total cost ($F=18.81,\ p<0.001$) than their low-performing counterparts.

5 Results

This section contains the results of training hidden Markov models on various segments of the behavioral study data. First, hidden Markov models were trained on the data from each study as a whole, providing an aggregate process model of designer activity. Next, separate hidden Markov models were trained on the data from high- and low-performing designers, providing an indication of beneficial heuristics.

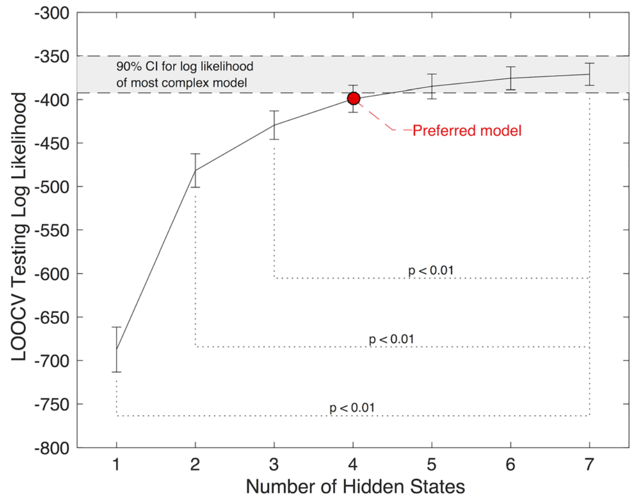

Fig. 4 Testing log-likelihood on truss study data for models with increasing number of hidden states (error bars show ±1 S.E.)

5.1 Aggregate Process Models.

Figure 4 shows the results of training hidden Markov models with varying numbers of hidden states on data from the truss study. Error bars indicate one standard error (S.E.). Models trained with three or fewer hidden states had testing log-likelihood values that were significantly lower than the most complex model (seven hidden states in this case), indicating that these models offered relatively low accuracy. The first model that is statistically indistinguishable from the highest complexity model in terms of testing log-likelihood is that trained with four hidden states. Therefore, $k=4$ is the best value to use in the current application, as it strikes a balance between parsimony and descriptive accuracy.

The transition and emission matrices for a four-state model trained across all truss study are provided in Fig. 5. The shading 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. The operation labels for the emission matrix also contain a percentage indicating the raw frequency with which each operation occurred.

The transition matrix of the hidden Markov model (shown on the left in Fig. 5) 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 each state are shown in the emission matrix on the right in Fig. 5. 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 Fig. 6 that indicates the most likely operations for each state. Circular nodes indicate the hidden states, numbered 1–4. 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 between hidden states (if the arrow connects two circular nodes) or the probability of a design operation given the current hidden state (if the arrow connects a circular node to a rectangular node).

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%

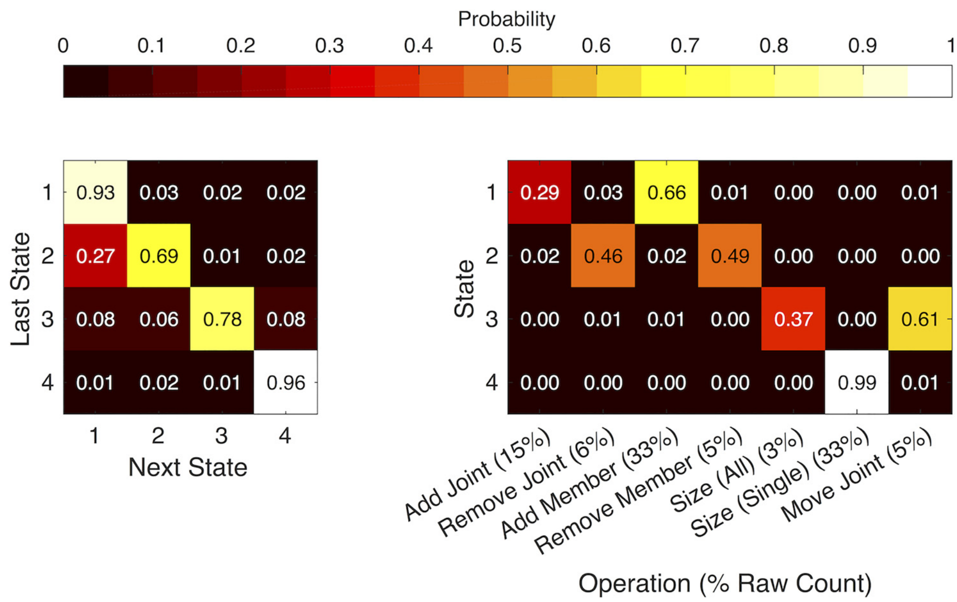

Fig. 5 Transition (left) and emission (right) matrices for four-state hidden Markov model based on the data from the truss study

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. This argument is supported by the relative infrequency of removal operation (see raw operation frequencies in Fig. 5).

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 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.

Figure 7 shows the results of training hidden Markov models with varying numbers of hidden states on data from the cooling system study. Again, the first model that is not statistically different from the most complex model has four hidden states. Therefore, $k = 4$ is the best value to use in the current application since it offers a balance between accuracy and simplicity. This is the same number of hidden states selected for the truss design data, which is surprising given that the behavioral studies were conducted with two significantly different design problems (structural versus fluid/thermal), and that the number of operations used per participant differed by nearly an order of magnitude (400–500 operations/participant for the truss problem versus 50 operations/participant for the cooling system problem). The commonality of four hidden states may be due to a common cognitive constraint that holds across different configuration design problems.

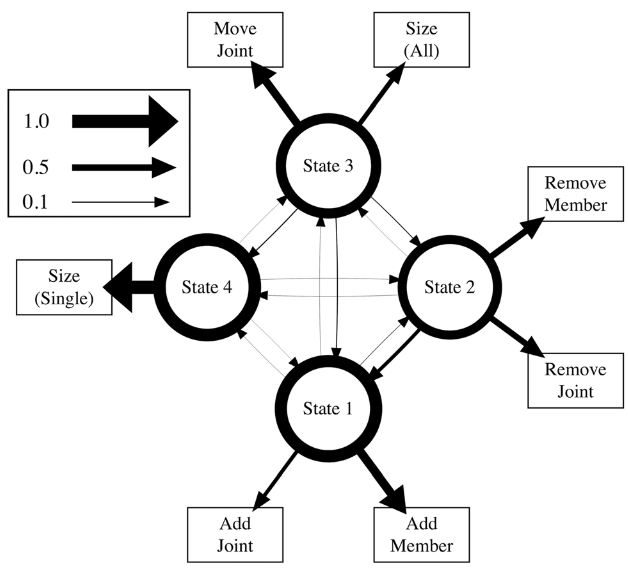

Fig. 6 Visual representation of four-state hidden Markov model based on data from truss study

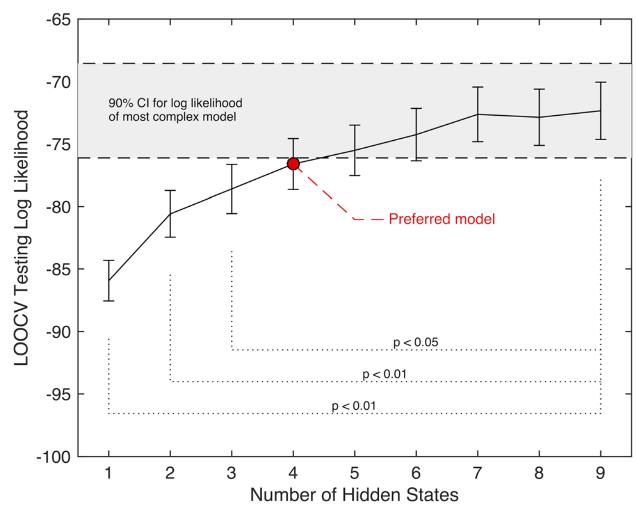

Fig. 7 Testing log-likelihood on cooling system study data for models with increasing number of hidden states (error bars show ±1 S.E.)

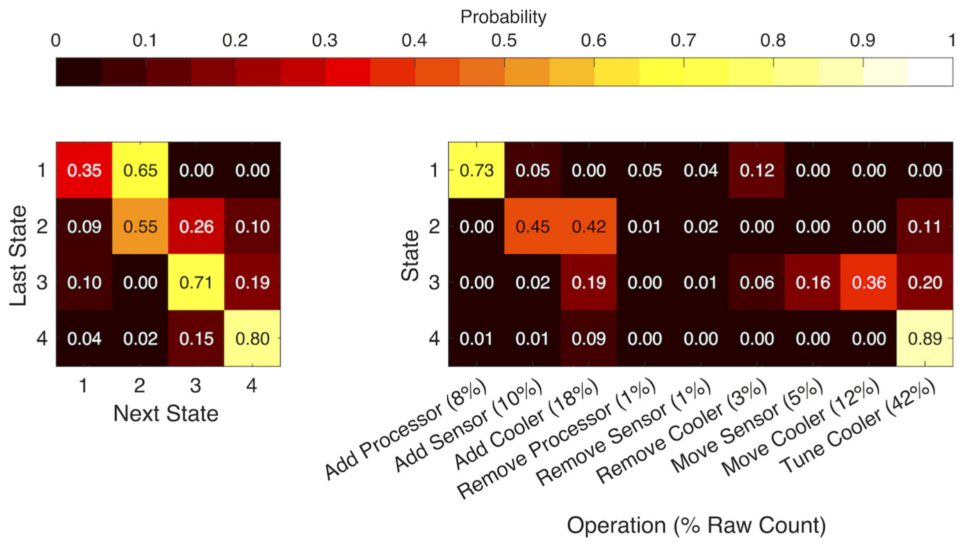

Fig. 8 Transition (left) and emission (right) matrices for four-state hidden Markov model based on the data from the cooling system study

The transition and emission matrices for a four-state model trained on all available data are provided in Fig. 8. In this model, the transition matrix has some strong diagonal elements, much like the preferred model for the truss study. However, unlike the truss study, 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. These observations show that designers in the cooling system study frequently moved between strategic states, whereas designers in the truss study tended to remain within a state for longer periods. A similar degree of operational restlessness is apparent in nonhidden Markov chain models that are trained on the same data [7].

A graph-based representation of the model is also provided in Fig. 9, which indicates the most likely operations for each state. As with the truss study, 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 the cooling system study 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, several 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.

Moreover, states 3 and 4 have a cyclic relationship with 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

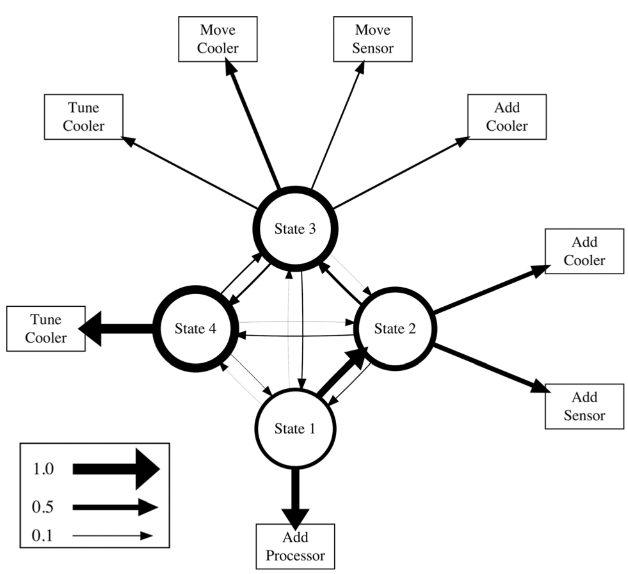

Fig. 9 Visual representation of four-state hidden Markov model based on the data from the cooling system study

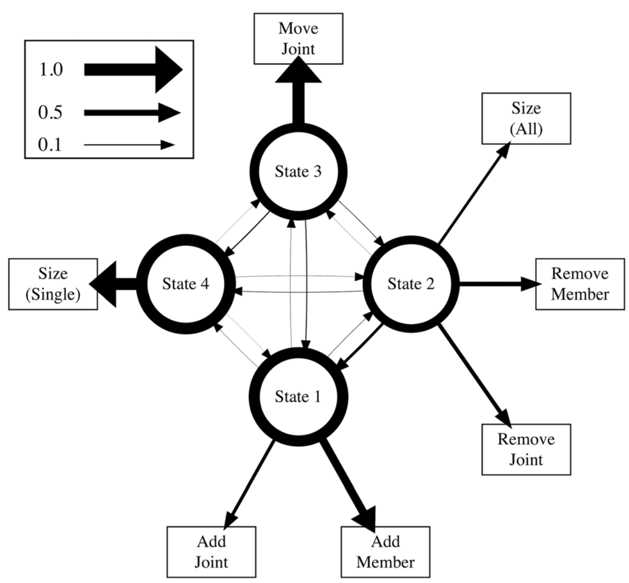

Fig. 10 Visual representation of four-state hidden Markov model for high-performing participants from the truss study. Model for low-performing participants is structurally identical to the aggregate model (Fig. 6).

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.

5.2 Performance-Differentiated Process Models. This section uses hidden Markov models to compare the high- and low-performing designers from each of the studies (see Sec. 4.3 for a description of how performance was designated). Hidden Markov models were trained using four hidden states, since this structure was identified in Sec. 5.1 as being capable of accurately describing design activity. The model trained on the low-performing segment from the truss study is structurally identical to the aggregate model from Sec. 5.1 (see Fig. 6). However, the model trained on the high-performing designers from that study (shown in Fig. 10) shows marked differences from both the aggregate model and the low-performing model. The transmission and emission matrices for the models trained on high- and low-performing data are provided in Figs. 11(*a*) and 11(*b*), respectively. These further illustrate the differences between the performance-differentiated models.

The primary structural difference in the model is that the operation for global resizing ("size (all)") shifts from state 3 to state 2. This indicates the introduction of a parameter operation to a state that was previously dominated by topology operations (namely, operations for removing elements from the truss), indicating that coarse member-sizing operations should be applied early during truss design to better guide the subsequent design operations. A secondary effect of this rearrangement is that state 3 is now devoted entirely to moving joints of the truss, which might indicate the importance of sweeping through a truss design to adjust joint locations.

The model trained on the low-performing segment from the cooling system study is also structurally identical to the aggregate model (see Fig. 9). However, the model trained on the high-performing data from that study (shown in Fig. 12) once again shows marked differences from both the aggregate model and the low-performing model. The transmission and emission matrices for the models trained on high- and low-performing data are provided in Figs. 13(*a*) and 13(*b*), respectively.

In this case, two distinct differences are apparent. First, the operation for tuning a cooler is moved from state 3 to state 2. This once again results in the introduction of a parameter operation to a previously topology-dominated state, while simultaneously resulting in a state that is completely devoted to spatial operations (in

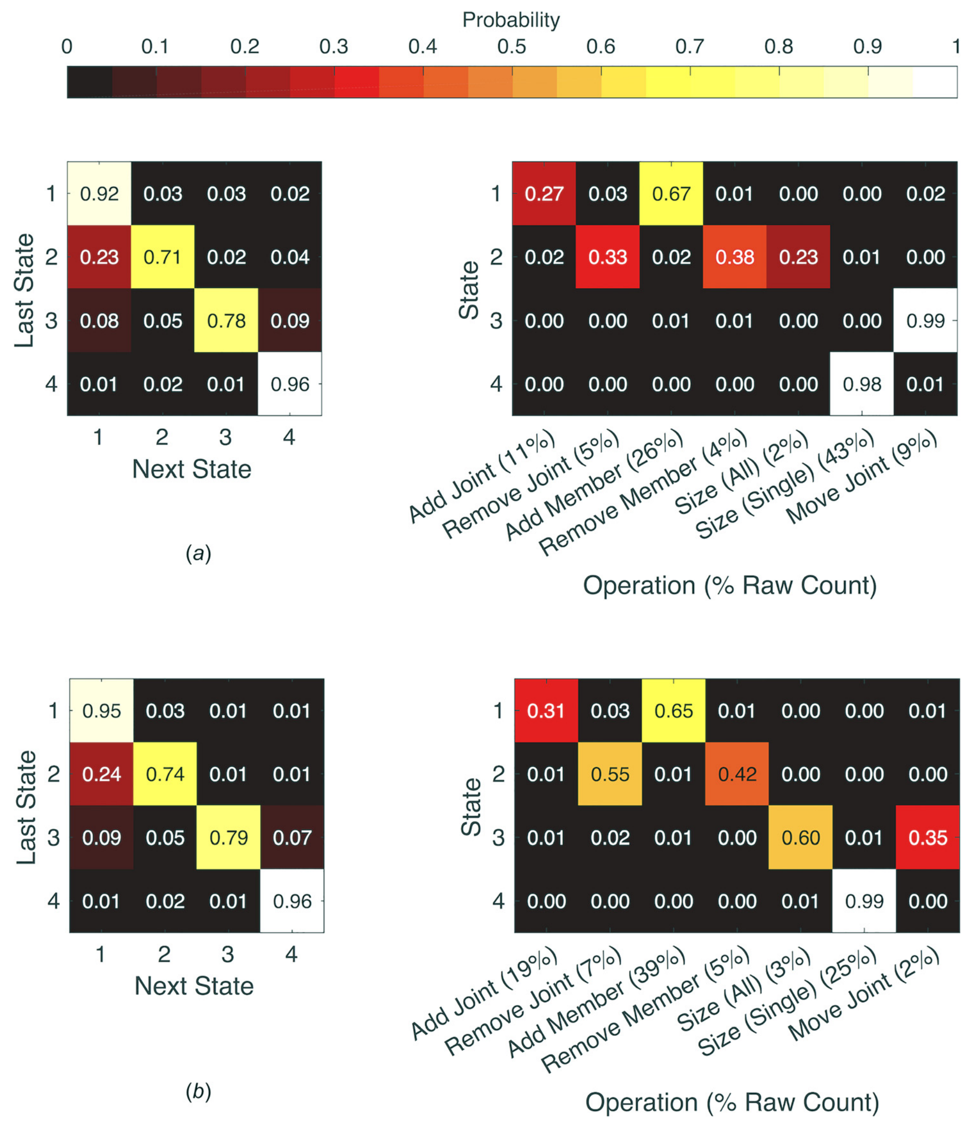

Fig. 11 Comparison of transmission/emission matrices for the truss study trained on: (*a*) high-performing data and (*b*) low-performing data

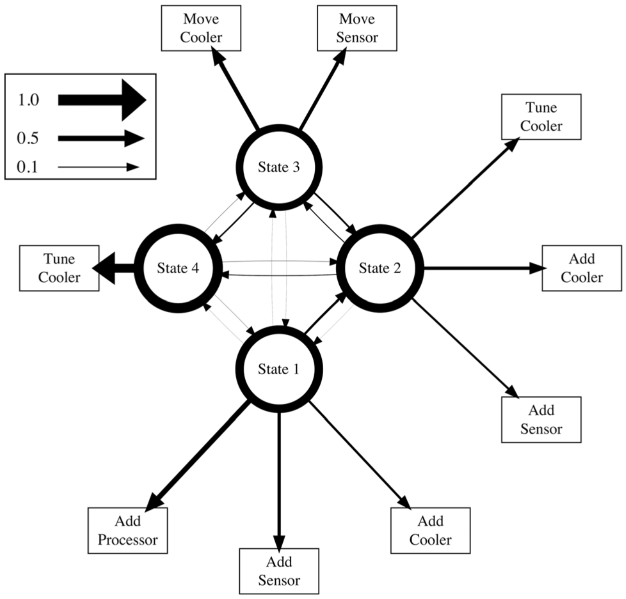

Fig. 12 Visual representation of four-state hidden Markov model for high-performing participants from the cooling systems design study. Model for low-performing participants is structurally identical to aggregate model (Fig. 9).

this case, moving coolers and sensors). The second change is the restructuring of topology operations in state 1. In the aggregate and low-performance models, state 1 is dominated by the operation for instantiating a new processor. However, in the high-performance model, state 1 is a more general topology addition state, containing operations for adding sensors and coolers in addition to processors. This may be evidence of a more incremental approach in high-performing participants. Whereas low-performing participants may have instantiated several processors (state 1) and then attached sensors and coolers to them (state 2), high-performing participants fully instantiate a minimal subsystem (one sensor, one cooler, and one processor) before moving to state 2 to elaborate the subsystem, or remaining in state 1 to add an additional minimal subsystem.

6 Discussion

There is substantial similarity between the various hidden Markov models produced for the truss study and the cooling system study. The first similarity is that the aggregate models selected for both studies contained four hidden states that were consistent in terms of operation type (topology, spatial, or parameter). A topology operation changes the connectivity of components in a solution (for instance, adding a new joint to a truss, or removing a sensor from a cooling system). A parameter operation changes the value of an attribute associated with a specific

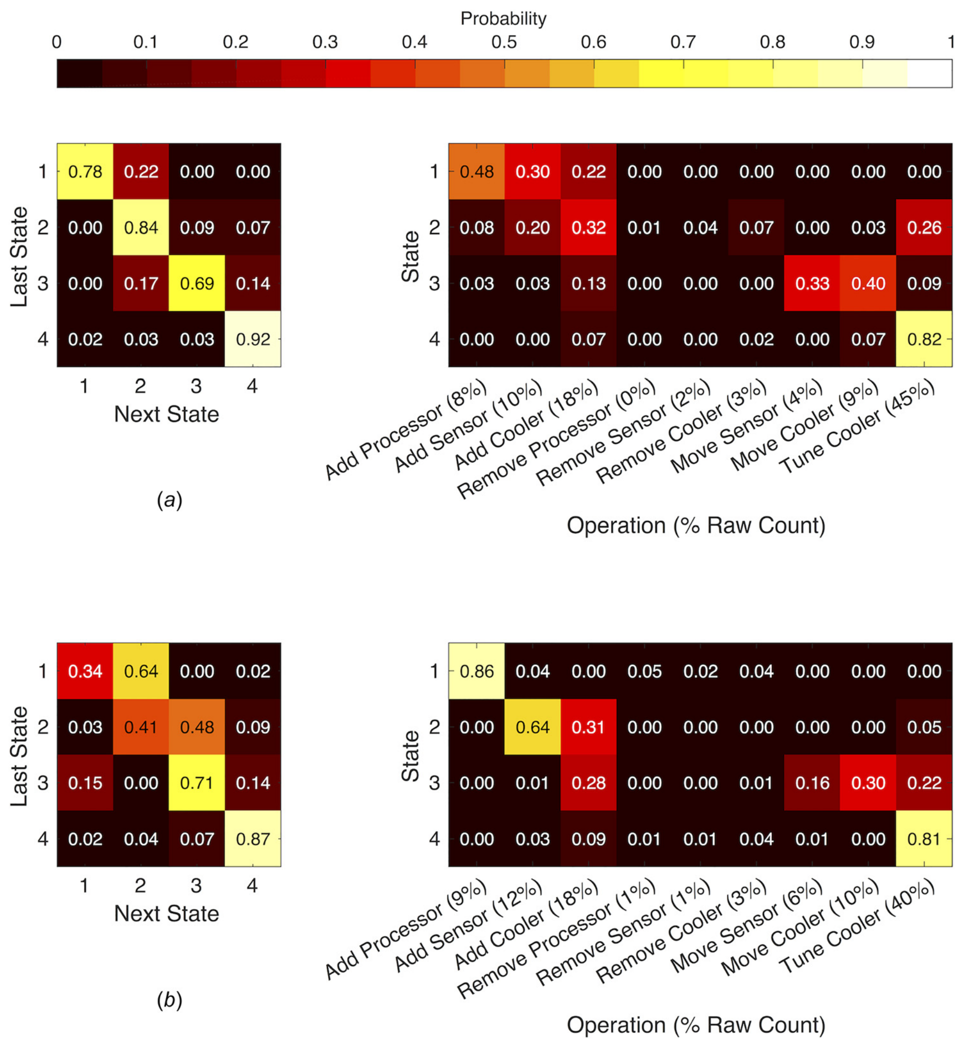

Fig. 13 Comparison of transmission/emission matrices for the cooling system study trained on: (*a*) high-performing data and (*b*) low-performing data

Table 2 Summary of states identified for aggregate models (identical to states of low-performing models)

	Truss study		Cooling system study	
State	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	Move sensor Move cooler Add cooler Tune cooler	Spatial Spatial Topology Parameter
4	Size (single)	Parameter	Tune cooler	Parameter

Table 3 Summary of states identified for high-performance models. The difference between this model and the aggregate/low-performance model is indicated by operations that are underscored (present for high-performing only) or struck through (present for low-performing only).

	Truss study		Cooling system study	
State	Operations	Type	Operations	Type
1	Add joint Add member	Topology Topology	Add processor <u>Add cooler</u> <u>Add sensor</u>	Topology <u>Topology</u> <u>Topology</u>
2	Remove joint Remove member <u>Size (all)</u>	Topology Topology <u>Parameter</u>	Add cooler Add sensor <u>Tune cooler</u>	Topology Topology <u>Parameter</u>
3	Move joint ~~Size (all)~~	Spatial ~~Parameter~~	Move sensor Move cooler ~~Add cooler~~ ~~Tune cooler~~	Spatial Spatial ~~Topology~~ ~~Parameter~~
4	Size (single)	Parameter	Tune cooler	Parameter

component of a solution (for instance, the cross section of a structural member or the flow rate of a cooler). Finally, spatial operations change the relative location of a component (for instance, moving a joint in a truss or moving a product in a cooling system). While spatial operations are technically a subset of parameter operations, we treat them distinctly because of the important role that spatial cognition plays in configuration design [8].

A summary of predominant operation types in each of the hidden Markov states from the aggregate models of the two studies is provided in Table 2. 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. The fourth and final state focuses purely on optimizing the parameters of the solution. These commonalities may be an indication of a common human approach to solving configuration design problems, a cognitive limit on human processing, or both. Interestingly, other work has found a four-state hidden Markov model to accurately depict cognitive states in some specific reasoning and information-processing tasks [23,32].

This work also used hidden Markov models to analyze the characteristics of high- and low-performing designers from the behavioral studies. The low-performance models are both structurally identical to the aggregate models of the same study (although the exact transition and emission parameter values differed slightly), while the high-performance models show distinctly different patterns of activity. In other words, the status quo pattern (followed by the majority of designers in both studies) resulted in solutions with mid-to-low quality, while achieving high quality required a departure from that majority. For both studies, the high-performance models (provided in Table 3) differ from the aggregate models in that a parameter operation shifts from state 3 to state 2. This has the effect of introducing a parameter operation early in the design process, in a state that was previously devoted solely to topology operations. A secondary effect of this rearrangement is that the third state in the high-performance models is devoted specifically to spatial operations.

The benefit derived from the differences between high- and low-performing models is likely twofold. First, the devotion of state 3 to spatial operations indicates that high-performing designers are engaged in prolonged phases of spatial activity with no modification of solution parameters. Since a spatial operation can impact multiple portions of the current solution, interweaving parameter operations with spatial operations (as did the low-performing designers) can be premature. A superior strategy, as evidenced by this analysis, is to engage in spatial and parameter optimization separately. Second, the incorporation of a parameter operation in an early state that is otherwise dominated by topology (namely, state 2) indicates that high-performing designers co-evolved solution topology and parameter values. Rather than focusing purely on modifying topology like the low-performing designers did, high-performing designers incorporated parameter tuning operations concurrently with topology operations, which allowed them to more accurately approximate the final performance of an early concept. This resulted in their ability to select a more effective topology, which in turn led to a higher quality solution. Initially, early application of parameter operations may appear to be greedy in nature, since the designer appears to be making needlessly small improvements at an early stage. However, the alternation between levels of detail (topological and parametric) is better described as opportunism [10]. Identifying it as such aligns this work with other research that has correlated opportunistic design activity with beneficial design outcomes [12]. This also implies that high-performing designers may have been responsive to emerging requirements or the attributes of current solutions, while low-performing designers were not. This result does not mean that designers should completely merge topology design and parametric design activities to achieve high performance, but rather that there is value in rapidly and opportunistically switching between the two early in design. In addition, the fact that both this work and that by Bender and Blessing [12] link opportunistic behavior to high quality solutions lends credence to the use of hidden Markov models for analyzing design process.

State 3 of the aggregate/low-performing model for the cooling system study (see Table 2) is also composed of several operation types at different levels of detail (topological, spatial, and parametric). While this may be indicative of rapid switching between operation types, and thus evidence of opportunism, state 3 of the corresponding truss model shows a different structure without topology operations. Thus, the presence of topology operations in state 3 of the cooling system model is likely a problem-specific anomaly, and not indicative of the broader, more general heuristics that are the target of this work.

The analyses on data from high- and low-performing participants are based on using four hidden states (a value chosen analytically in Sec. 5.1). For models trained on fewer hidden states (three or less), the models become so general that the structure of models trained on high- and low-performing data becomes nearly identical. Using a greater number of hidden states (five or more), increases the differences between the models trained on high- and low-performance data, but these differences are so multifarious that concise interpretation becomes challenging. Using exactly four-hidden states makes it possible to resolve models that differentiate between performance levels, while sufficiently generalizing the differences so that they become readily interpretable. This value was selected using a quantitative procedure.

In sum, this work demonstrates that high-performing designers engage in early search for an adequate system topology by estimating final quality through the application of a limited number of parameter operations concurrently with topology operations. This approach contrasts with low-performing designers, who almost perfectly separate topology and parameter operations. The heuristic approach of performing some parameter operations in concert with topology design has the potential to inform education and training of designers, through instruction in this strategy as a means of increasing availability of information during early search. There is also the potential to use this heuristic to inform more intelligent design algorithms, perhaps influencing the probabilistic order in which different types of design operations are applied over the course of solving.

7 Conclusions

This work used a data-mining approach to identify heuristics and patterns of activity displayed by designers in two behavioral studies. One of the behavioral studies challenged participants with the design of a truss structure, while the other tasked them with the design of an internet-connected home cooling system. The tasks in both studies can be classified as cases of full configuration design. A full record of the operations performed by participants during solving was recorded, and formed the corpus of data that was mined with hidden Markov models.

The first portion of this work applied hidden Markov models to the entirety of the data from each behavioral study, producing aggregate process models of designer activity. These models revealed strong similarities between the designers' approaches to the two studies, even though the tasks used in the studies were drawn from different domains (structural versus thermal/fluid) and that participants in one study used nearly ten times more operations than participants in the other. The first similarity was that the best model for each study had four hidden states. Further, the specific states from the two models matched in terms of the type of operation in each: the first two states were devoted to topology operations, the third state contained spatial (and other) operations, and the final state was devoted to parameter operations. The high degree of similarity between the two models indicates that these state patterns may be generalizable beyond the studies examined here across the broader class of configuration design problems.

The second portion of this work leveraged hidden Markov models to compare the processes of high- and low-performing designers. For each study a group of high-performing designers and a group of low-performing designers were designated, and a separate hidden Markov model was trained on each of these groups. There was once again alignment between the models from the two studies. The models of low-performing participants were identical in structure to the corresponding aggregate models, while the high-performing models showed different patterns of activity. The key difference was that high-performing designers incorporated parameter operations in states that were otherwise dominated by topology operations, a pattern that may be indicative of opportunistic design behavior. In doing so, high-performing designers could roughly tune the parameters of early design concepts. This in turn allowed them to better estimate the final quality of early concepts, providing more accurate information to guide the search process.

This work showed that hidden Markov models can be an effective tool for describing design processes. By employing this statistical tool, it was demonstrated that nuanced early search is crucial to creating high-quality configurations, and that more effective early search can be obtained through the judicious application of parameter operations concurrently with topology design. Three directions of future inquiry stem from the current work. First, an immediate avenue of future work should experimentally validate this heuristic and variants of it as a means of both monitoring progress toward a solution and improving the quality of designers' solutions. This should be undertaken experimentally using well-defined problems such as those investigated here as well as problems with ill-defined objective functions. The latter has the potential to definitively demonstrate the utility of the identified heuristics for use by human designers. Second, the benefit of the identified heuristic of applying early parameter operations should be examined for use in design synthesis algorithms. Often, design algorithms separate topology design from parameter design—this work shows that there may be some benefit from partially merging the two. Finally, this work identified a four-state model of design process based on the analysis of human behavior, which aligns with research that has identified four-state process models in other tasks [23,32]. Fundamental investigation is necessary to determine if this number is a meaningful cognitive constant or constraint that holds across a variety of problems.

Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors. A preliminary version of part of this work was published in the proceedings of the 2017 International Conference on Engineering Design [33].

Funding Data

- Air Force Office of Scientific Research (Grant No. FA9550-16-1-0049).
- Division of Graduate Education (Grant No. DGE125252).

References

[1] Mittal, S., and Frayman, F., 1989, "Towards a Generic Model of Configuration Tasks," 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, MI, Aug. 20–26, Vol. 2, pp. 1395–1401.
[2] 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," Des. Stud., **36**(1), pp. 99–121.
[3] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Optimizing Design Teams Based on Problem Properties: Computational Team Simulations and an Applied Empirical Test," ASME J. Mech. Des., **139**(4), p. 041101.
[4] Chandrasekaran, B., 1990, "Design Problem Solving: A Task Analysis," AI Mag., **11**(4), pp. 59–71.
[5] Wielinga, B., and Schreiber, G., 1997, "Configuration-Design Problem Solving," IEEE Expert Intell. Syst. Appl., **12**(2), pp. 49–56.
[6] McComb, C., Cagan, J., and Kotovsky, K., 2015, "Lifting the Veil: Drawing Insights About Design Teams From a Cognitively-Inspired Computational Model," Des. Stud., **40**, pp. 119–142.
[7] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Capturing Human Sequence-Learning Abilities in Configuration Design Tasks Through Markov Chains," ASME J. Mech. Des., **139**(9), p. 091101.
[8] Kim, M. J., and Maher, M. L., 2008, "The Impact of Tangible User Interfaces on Spatial Cognition During Collaborative Design," Des. Stud., **29**(3), pp. 222–253.
[9] Snider, C., Dekoninck, E. A., and Culley, S. J., 2014, "The Appearance of Creative Behavior in Later Stage Design Processes," Int. J. Des. Creativity Innovation, **2**(1), pp. 1–19.
[10] Visser, W., 1994, "Organisation of Design Activities: Opportunistic, With Hierarchical Episodes," Interact. Comput., **6**(3), pp. 239–274.
[11] Guindon, R., 1990, "Designing the Design Process: Exploiting Opportunistic Thoughts," Hum.-Comput. Interact., **5**(2), pp. 305–344.
[12] Bender, B., and Blessing, L., 2004, "On the Superiority of Opportunistic Design Strategies During Early Embodiment Design," Eighth International Design Conference (DESIGN), Dubrovnik, Croatia, May 18–21, pp. 1–6.
[13] Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G., 1998, *Biological Sequence Analysis*, Cambridge University Press, Cambridge, UK.
[14] Baum, L. E., and Petrie, T., 1966, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," Ann. Math. Stat., **37**(6), pp. 1554–1563.
[15] 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," Bull. Am. Math. Soc., **73**(3), pp. 360–364.
[16] Baum, L. E., and Sell, G., 1968, "Growth Transformations for Functions on Manifolds," Pac. J. Math., **27**(2), pp. 211–227.
[17] 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," Ann. Math. Stat., **41**(1), pp. 164–171.
[18] Jelinek, F., Bahl, L., and Mercer, R., 1975, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," IEEE Trans. Inf. Theory, **21**(3), pp. 250–256.
[19] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D., 1994, "Hidden Markov Models in Computational Biology: Applications to Protein Modeling," J. Mol. Biol., **235**(5), pp. 1501–1531.

[20] Gonzalez, A. M., Roque, A. M. S., and Garcia-Gonzalez, J., 2005, "Modeling and Forecasting Electricity Prices With Input/Output Hidden Markov Models," IEEE Trans. Power Syst., **20**(1), pp. 13–24.
[21] White, B., Blaylock, N., and Bölöni, L., 2009, "Analyzing Team Actions With Cascading HMM," The Florida Artificial Intelligence Research Society Conference (FLAIRS), Sanibel Island, FL, May 19–21, pp. 129–134.
[22] Tenison, C., and Anderson, J. R., 2016, "Modeling the Distinct Phases of Skill Acquisition," J. Exp. Psychol. Learn. Mem. Cognit., **42**(5), pp. 749–767.
[23] Anderson, J. R., and Fincham, J. M., 2014, "Discovering the Sequential Structure of Thought," Cognit. Sci., **38**(2), pp. 322–352.
[24] 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," J. Eng. Educ., **96**(4), pp. 359–379.
[25] Stempfle, J., and Badke-Schaub, P., 2002, "Thinking in Design Teams—An Analysis of Team Communication," Des. Stud., **23**(5), pp. 473–496.
[26] Goldschmidt, G., and Rodgers, P. A., 2013, "The Design Thinking Approaches of Three Different Groups of Designers Based on Self-Reports," Des. Stud., **34**(4), pp. 454–471.
[27] Kotovsky, K., Hayes, J., and Simon, H. A., 1985, "Why Are Some Problems Hard? Evidence From Tower of Hanoi," Cognit. Psychol., **17**(2), pp. 248–294.
[28] McComb, C., Jonathan, C., and Kotovsky, K., 2017, "Validating a Tool for Predicting Problem-Specific Optimized Team Characteristics," ASME Paper No. DETC2017-67430.
[29] Dempster, A., Laird, N. M., and Rubin, D. B., 1977, "Maximum Likelihood From Incomplete Data Via the EM Algorithm," J. R. Stat. Soc.: Ser. B, **39**(1), pp. 1–38.
[30] Stratonovich, R. L., 1960, "Conditional Markov Processes," Theory Probab. Its Appl., **5**(2), pp. 156–178.
[31] Arlot, S., and Celisse, A., 2010, "A Survey of Cross-Validation Procedures for Model Selection," Stat. Surv., **4**, pp. 40–79.
[32] Borst, J. P., and Anderson, J. R., 2013, "Discovering Processing Stages by Combining EEG With Hidden Markov Models," 35th Annual Conference of the Cognitive Science Society (CogSci), Berlin, July 31–Aug. 3, pp. 221–226.
[33] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Eliciting Configuration Design Heuristics With Hidden Markov Models," International Conference on Engineering Design (ICED), Vancouver, BC, Canada, Aug. 21–25, Paper No. 162.