

Capturing Human Sequence-Learning Abilities in Configuration Design Tasks Through Markov Chains

Christopher McComb

Mem. ASME
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: com@cmu.edu

Jonathan Cagan

Fellow ASME
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: cagan@cmu.edu

Kenneth Kotovsky

Department of Psychology,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: kotovsky@cmu.edu

Designers often search for new solutions by iteratively adapting a current design. By engaging in this search, designers not only improve solution quality but also begin to learn what operational patterns might improve the solution in future iterations. Previous work in psychology has demonstrated that humans can fluently and adeptly learn short operational sequences that aid problem-solving. This paper explores how designers learn and employ sequences within the realm of engineering design. Specifically, this work analyzes behavioral patterns in two human studies in which participants solved configuration design problems. Behavioral data from the two studies are first analyzed using Markov chains to determine how much representation complexity is necessary to quantify the sequential patterns that designers employ during solving. It is discovered that first-order Markov chains are capable of accurately representing designers' sequences. Next, the ability to learn first-order sequences is implemented in an agent-based modeling framework to assess the performance implications of sequence-learning abilities. These computational studies confirm the assumption that the ability to learn sequences is beneficial to designers. [DOI: 10.1115/1.4037185]

1 Introduction

Designers often search for new solutions by iteratively adapting a current design. By engaging in this search, designers progressively improve the quality of their solutions. However, they also begin to learn what operational patterns are likely to improve the solution in future iterations. The current work examines designers' capacity for learning and applying beneficial operation sequences, and studies the impact of such behavior on performance.

The current work specifically stems from observations of small teams of engineering students engaged in the design of a truss [1]. In a subsequent analysis of that study, it was hypothesized that the precise order in which operations were performed may have impacted the quality of solutions [2]. The analysis in the current work focuses expressly on these operational sequences and is thus conducted at shorter timescales and at a finer resolution than other research that has studied the sequencing of design stages or design tasks (which is reviewed in Sec. 2). This is the level at which designers and engineers explicitly engage in their iterative search for solutions, so choosing the best actions becomes of paramount concern for the creation of high-quality solutions. This work is specifically centered on two overarching questions:

- (1) *How much representational complexity is necessary to quantify the sequential patterns that designers employ during solving?* Previous work has used sequential models with varying degrees of complexity to examine designer activity [3–6]. However, no direct analysis has been conducted to assess what degree of complexity is necessary to offer an accurate aggregate representation of designer activity. The current work utilizes Markov chain concepts to verify the fundamental assumption that sequential treatments are necessary, and to uncover the necessary level of complexity.
- (2) *Does the use of operation sequences benefit designers?* Effective design must find a balance between exploration

of a design space and exploitation of known features of the design space to achieve a solution [7–9]. Sequence learning may serve to augment exploitation in design, similar to the role that it plays for solving puzzle problems [10,11]. However, it is also possible that this augmentation occurs at the expense of effective exploration, as designers may apply learned sequences to greedily improve solution quality rather than searching broadly for solution alternatives. Studying the performance implications of sequence learning is complicated by the fact that sequence learning can take place implicitly [12,13], which makes it challenging to control, observe, and assess. This work equips a computational model of engineering design teams with Markovian constructs to accurately assess the performance implications of sequence-learning abilities.

These research questions are addressed by using Markov chain constructs to represent and simulate the sequential pattern of human behavior in design. While Markov chains do not extract finite operation sequences, they do implicitly represent such sequences using probabilistic chains. The mathematical underpinnings of Markov chains are described in greater detail in Sec. 2, along with relevant information pertaining to sequencing and design.

This paper is comprised of two investigations exploring each of the overarching research questions. The first explores the degree of complexity necessary to accurately characterize the sequences used by designers. This is accomplished by applying a statistical analysis to the human data from two previously conducted cognitive studies. This analysis reveals that participants in both studies employed sequences of operations when constructing solutions. The results also show that operation sequencing in both studies can be characterized as a first-order Markov process. Higher-order Markov models display accuracy that is statistically equivalent to first-order models. The second investigation attempts to assess the performance implications of operation sequencing. This assessment is attained by computationally simulating the activities of design teams using the Cognitively Inspired Simulated Annealing Teams (CISAT) modeling framework, an agent-based platform that has been shown to approximate the process and performance

Contributed by the Design Theory and Methodology Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received November 14, 2016; final manuscript received June 16, 2017; published online July 12, 2017. Assoc. Editor: Katja Holttä-Otto.

characteristics of engineering design teams [2]. The insight that human operation sequencing can be treated as a first-order Markov process is used to equip CISAT agents with sequence-learning abilities, enabling a computational comparison between teams with and without the ability to learn sequences. These simulations demonstrate that sequence-learning abilities were helpful to designers in the cognitive studies, and that similar computational implementations may be of use for automated design synthesis.

2 Background

The patterns that humans identify through sequence learning are essential to the execution of both mundane and specialized tasks [14]. Patterns are also identified through chunking, a related behavior in which humans assemble many pieces of related information concomitantly in memory [15–17]. Chunking behavior has even been mirrored in computational design algorithms [18]. These two behaviors are differentiated by the modality of the recognized patterns—chunking extracts patterns that are spatial or relational, while sequencing extracts temporal patterns. Both are important to design cognition, but the current works focus on the latter.

2.1 Sequence Learning. Sequential behavior can be an indicator of expertise in some domains [19]. However, it has also been shown that participants are capable of quickly acquiring and employing move sequences [10,11,20]. Participants solving the Tower of Hanoi puzzle spend most of their time learning to compose appropriate sequences (namely, pairs) of moves [10]. Once they learn how to do so, they spend a relatively short amount of time to actually solve the puzzle [10]. Further, a comparison of several isomorphs of the Tower of Hanoi puzzle revealed that isomorph difficulty increased time spent in the learning phase, but the time spent in the solving phase was invariant with respect to isomorph difficulty [10]. In studies using the Thurstone letter series completion task, two procedural steps were identified in participants' problem-solving efforts [20]. The first step entails identifying some structure in the letter series and creating a rule that describes it, and the second involves leveraging that rule to extrapolate the next letters in the series. These steps are abbreviated as *generating a pattern* and *generating a sequence*. This process has been reevaluated and confirmed with both computational simulations and cognitive studies [20,21]. Other work has shown that participants preferred specific operation orders in solving geometric analogy tasks, despite the fact that the task itself did not place explicit constraints on the permissible order of operations [22]. Participants performed with lower accuracy and speed when made to use a nonpreferred order, indicating that appropriate sequencing of operations has strong implications for performance [22].

Several studies have shown that humans can learn sequences implicitly (i.e., without direct attention) [12,13]. However, studies have also shown that direct attention while learning sequences can boost positive outcomes [23,24]. Together, these findings underscore the existence of two alternative pathways through which sequence learning can occur [14,25].

2.2 Sequencing in Design. The role of sequencing as it pertains to design has been examined with respect to stages, tasks, and operations. These three sequence types can be conceptualized along a spectrum of abstraction, from design stages (the most abstract and general) to design operations (the least abstract and most detail-specific). A similar continuum can be constructed to describe the timescale at which these objects are enacted, with design stages being enacted at longer timescales, and design operations typically at shorter timescales.

Observations of individual designers (or of design teams) are typically used to study the sequencing of design stages. One study coded design team communication according to alignment with design stages [26]. It was discovered that design teams were likely

to focus their discussion on a specific stage for several utterances before transitioning to other stages [26]. In another study, participants were tasked with designing a playground and their activities were again coded according to alignment with design stages [27]. The procedural sequences exhibited by experts tended to transition smoothly and linearly between stages, while sequences exhibited by novices were more erratic, with frequent stage changes [27]. It has also been shown that there is substantial variability in the order in which both Ph.D. and undergraduate students employ design stages, with few participants transitioning linearly between the stages [28]. A similar study demonstrated that transitioning linearly through the design process tended to produce solutions of higher quality [29].

The design tasks are typically enacted at shorter timescales than design stages. Appropriate ordering of design tasks can increase the concurrency with which tasks can be completed [30], decrease the time and cost involved in developing a product [31], and increase the information that is available for key design decisions [32]. Waldron and Waldron analyzed the sequencing of tasks observed during the design of an intricate mechanical system [33]. Their analysis showed that there is not always a clean break between conceptual and detailed design, due in part to the fact that tasks may carry over between design stages [33]. Theoretical research on task sequencing has demonstrated a possible link between problem complexity and optimal approaches for task ordering [32]. Other work has implemented genetic algorithms for optimizing task sequences with respect to a number of different objectives [34].

The sequencing of discrete design operations takes place on the shortest timescales and has the most intimate impact on potential solutions. This is the level at which designers and engineers explicitly engage in their iterative search for solutions, so choosing the best actions becomes of paramount concern 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 [35,36] are commonly used to create coding schemes to study sequencing at this scale. The FBS design ontology specifically describes the design as a process with the ultimate goal of transforming a set of design requirements into a design description [35]. The description cannot proceed directly from the set of requirements, but instead arises as a result of considering a number of 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. The first-order sequential behavior in the FBS ontology (the transitions between issues) has been modeled via Markov chains [3–5]. The second-order sequential behavior (the probability that specific processes will precede specific issues) has also been investigated [37]. Simulations have also considered the effects of memory on sequencing behavior in computational agents using higher-order Markov chains [6]. Aside from studies of human designers, the extraction and implementation of beneficial rule pairs have been explored with respect to design automation [38].

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 quantifiable effect on the form of the current design solution. Markov chain concepts are used to study the sequencing of operations for design tasks and also to implement sequence-learning abilities within CISAT, a computational model of design teams [2].

2.3 Markov Processes. A Markov chain is a mathematical model of a stochastic system that transitions between a set number of possible states [39]. Specifically, a first-order Markov chain assumes that the probability of transitioning to a future state

depends only on the current state of the system, and not on previous states [39]. These transition probabilities are stored in the transition matrix, T , where the value of T_{ij} is the probability of transitioning from state i to state j . The mathematics governing Markov chains were proposed in 1907 [40], and over the last century, Markov chains have been leveraged for computer performance evaluation [41], web search [42], modeling chemical processes [43], and analyzing design team communications [44,45].

Figure 1 gives an example of a first-order Markov chain with three states (S_1 , S_2 , and S_3). Arrows in the figure indicate possible transitions between states, and these are labeled with the corresponding element of the transition matrix. It should be noted that Markov chains typically permit self-transitions, meaning that the system fails to transition out of the current state for one or more time steps. In the current work, Markov chains describe the order in which study participants applied operations while constructing solutions. These modifying operations are modeled as the states of the Markov chain model, making the transition matrix a probabilistic description of operation sequences.

The higher-order Markov chains can also be implemented. In these models, the selection of the next state *does* depend on past states, thus modeling a degree of “memory” within the system [46]. In the context of design, the implicit memory of higher-order Markov chains could serve as a useful analog for a portion of the expertise and memory of human designers. The higher-order Markov chain models are used in this work to characterize how much inherent memory is necessary to specify the order in which study participants applied operations while constructing solutions. The zero-order Markov chains are also used in this work. These models do not encode a sequential representation of data, but instead encode the unconditional frequency with which operations are applied (much like the probabilities associated with each side of a weighted die).

3 Datasets

The operation datasets analyzed in this work were derived from two previously conducted cognitive studies. The first study tasked engineering students with designing a truss and was originally created to examine design in the face of dynamic problems [1]. The second study tasked a different group of engineering students with the design of an internet-connected home cooling system and was originally designed to assess team coordination and communication [47]. Neither study was designed explicitly for the analyses applied in this paper—rather, the current work mines patterns of human behavior from those preexisting datasets. In addition, the differences between the two studies are an advantage to the current work because the respective data provide a broad basis from which to draw more general conclusions. A brief review of both studies is given in this section, with a summary of important differences provided in Table 1. The disparate domains of the two

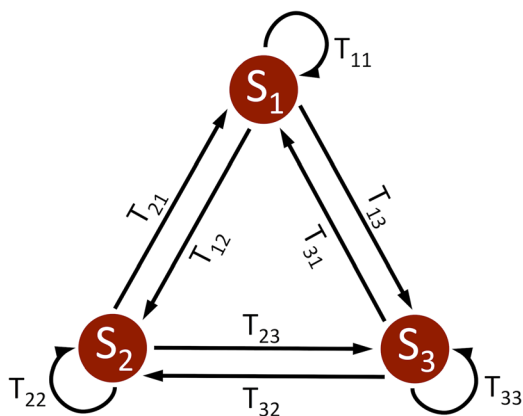


Fig. 1 Example Markov chain with three states

studies add to the generalizability of the results of this work, as does the varying number of operation types. Further, participants in the truss design study performed nearly an order of magnitude more operations than the participants in the home cooling system design study. Given these substantial differences, any similarities noted in human behavior between these two studies may be generalizable to the broader class of configuration design problems.

While the data used here were collected from *team*-based experiments, the focus of the current work is on *individual* sequence learning. 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 an aggregate collection of data at the team level. Second, in both studies, the time spent working individually was much larger than that spent interacting with teammates, so learned sequences were largely the result of individual activity and effort.

3.1 Truss Design Study. In this previously conducted study, 16 teams of three mechanical engineering students were tasked with designing a structural truss. The design was conducted over the course of six 4-min design sessions. New problem statements were introduced twice, without warning, in order to study problem-solving and design in response to a dynamic design task [1]. In the original problem statement, teams were instructed to design a truss to support a given load at the middle of each of two spans. The first change presented participants with the same general layout, but they were also instructed to account for the removal of one of the supports at any time. The second change instructed teams to design their truss to avoid an obstacle. Teams were given a separate target mass and factor of safety for each of the three problem statements. Over the course of the study, participants were permitted to interact freely with members of their team [1]. Estimates of communication frequency made in previous work vary from one interaction for every 30 individual actions to once for every 100 actions, depending on the team [2]. Because the problem statement changed during design, it is expected that this dataset will yield sequencing information that applies generally to a variety of truss design problems.

Every participant was also provided with a computer program that allowed them to construct, evaluate, and share design solutions with their teammates. This program also recorded the operations that participants selected while creating their designs which made it possible to reconstruct a full log of design activity. The allowed operations were as follows:

- (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, and
- (7) moving a joint.

The information generated by participants was analyzed following the experiment to produce a sequence of move operators (denoted by the integers 1–7) for every participant. Every sequence consisted of 400–500 operations. A short example operation sequence is depicted in Fig. 2.

3.2 Home Cooling System Design Study. This study 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). The design was conducted over the course of a 30-min session. Several experimental

Table 1 Summary of key differences between studies

	Truss design study	Home cooling system design study
Problem domain	Structural	Thermal/fluid
Number of distinct operation types	7	9
Average number of operations per participant	430.1 ± 123.5	50 ± 0 (fixed number)

conditions were established to control the frequency with which participants interacted with their teammates (from zero interaction to interacting once for every five individual actions). To ensure a common basis for comparison between conditions, every participant was allowed to perform only 50 design operations.

Every participant was provided with a program that allowed them to build, assess, and share solutions. It was also used to continuously record the operations that the participants used, much like the design program for the truss study. The operations available to participants here were as follows:

- (1) add processor,
- (2) add sensor,
- (3) add cooler,
- (4) remove processor,
- (5) remove sensor,
- (6) remove cooler,
- (7) move sensor,
- (8) move cooler, and
- (9) tune cooler.

This information was processed after the experiment to produce a list of move operators (denoted by the integers 1–9) for each of the 54 participants in the study. Every solution sequence consisted of exactly 50 operations. A short example sequence is provided in Fig. 3. The solution diagrams in the left column depict a plan view of the structure, with shading indicating the relative temperature in different rooms.

4 Investigation 1: Representation of Operation Sequences

This paper first analyzes human operation data from the two design studies with the objective of determining what degree of representational complexity is necessary to accurately model the sequences employed by designers.

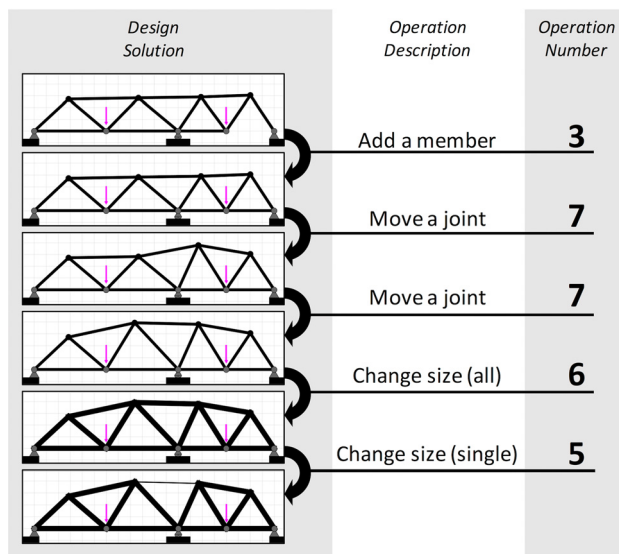


Fig. 2 Example truss operation sequences, with numbers corresponding to the list of truss operations

4.1 Methodology. Markov chains were trained on data from the design studies in order to provide a statistical representation of the sequence in which operations occurred. The following discussion of the process for training these models is based on material in Ref. [39], but is presented in terms of design operations (instead of Markovian states) to aid understanding of its relevance to the current work. 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 the elements in its transition matrix, 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 . The values in the transition matrix can be estimated based on observed data by computing

$$T_{ij} = \frac{N_{ij}}{N_i} \quad (1)$$

where N_{ij} is the number of instances in which operation j is observed to follow operation i , and N_i is the number of instances in which operation i is observed in total. The diagonal of the transition matrix contains probabilities for cases where $i = j$, indicating that an operation is followed by itself.

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

$$\mathcal{L}_{MC} = \sum_{i=1}^M \sum_{j=1}^M N_{ij} \cdot \ln(T_{ij}) \quad (2)$$

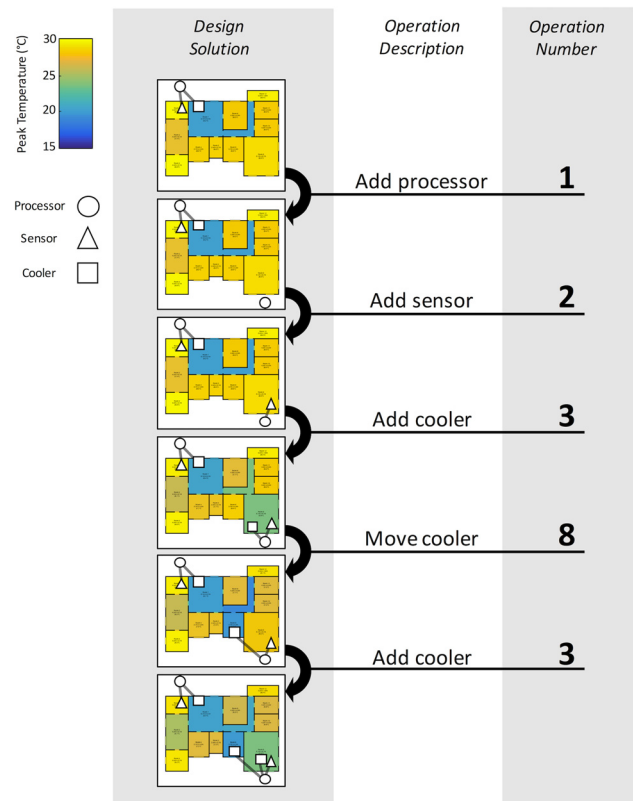


Fig. 3 Example cooling system operation sequence, with numbers corresponding to the list of system operations and shading indicating room temperatures

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

The procedure for training the higher-order Markov chains follows essentially the same pattern as that for the 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 datasets, 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 [48]. For a dataset 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 sample has been used for testing, providing n evaluations of the testing accuracy, for which the mean and standard error can be computed. It should be noted that leave-one-out cross-validation is a special case of k -folds cross-validation [49] for which k is equal to the number of samples (n). Using $k = n$ provides an accurate estimate with lower bias and a more conservative variance than values of $k < n$ [50].

In this work, each sample is composed of the data from one study participant (consisting of many operations). Thus, the validation approach used here estimates how accurate the model might be for describing the behavior of a previously unseen individual. It should be noted that during training (and for communicating final results), the transition probabilities are computed using the data from multiple study participants.

4.2 Results for Truss Design Study. A plot of log-likelihood for models of increasing order is shown in Fig. 4(a) with error bars indicating standard error. The dashed line shows the log-likelihood of the model on the training dataset, and the solid line shows the log-likelihood on the testing dataset. 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 4(a) 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 the higher-order models. In other words, the higher-order models begin to fit attributes of the training data that are not general, and thus exhibit lower accuracy on the testing dataset. There is a slight increase in the mean testing accuracy between the first-order and second-order models, but this increase is not significant ($F = 0.31$, $p > 0.5$).

As noted previously, the testing log-likelihood in Fig. 4(a) plateaus after the first-order model, with no further significant differences apparent in the testing log-likelihood curve. Therefore, the first-order model is preferred, as it provides a degree of accuracy that is statistically equivalent to the higher-order models, but it does so with much less complexity. Designer activity in the truss design task can, therefore, be accurately modeled as a first-order Markov process.

Comparing the zero-order model (which encodes a nonsequential representation) to the first-order model (which encodes a representation of designer activity that is both parsimonious 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

Fig. 4(c), and the transition matrix of the first-order Markov model is shown in Fig. 4(b). The shading inside the squares indicates the magnitude of the probability, which is also displayed numerically within each square.

A comparison of the statistical models provided in Figs. 4(b) and 4(c) justifies the substantially higher likelihood of the first-order model. The transition probability matrix of the first-order model has strong diagonal elements, which indicates that elements were fairly likely to be applied multiple times in a row. This type of sequential probabilistic dependence simply cannot be represented in a zero-order model. For example, consider the 33% chance that the next operation chosen by the zero-order model will be to add a member. Because of the assumptions of the model, this probability is not dependent on the previous action. However, the first-order model demonstrates that the choice to add a member is heavily dependent on what the previous operation was, and is particularly likely after adding a joint, removing a joint, or adding a member. Conversely, it is extremely unlikely to be chosen after changing the size of truss members. As another example, the zero-order model also contains a 33% chance that the next operation chosen will be to change the size of a single member. However, the first-order model provides a caveat with this value, showing that this operation is most likely to follow itself, and unlikely to occur after adding a joint, removing a joint, adding a member, or removing a member.

A graph-based visualization of the state and transition probabilities of the first-order Markov chain is provided in Fig. 5. Arrows are used to indicate transitions between states, and line thicknesses represent the relative probability of those transitions, with thicker lines indicating transitions with higher probability. This visualization only includes the transitions with the highest probabilities (specifically transitions with probabilities above the median, approximately 0.03). The self-transition probabilities are indicated by the border thickness of the circle around that operation. This figure helps to expose additional patterns of sequential action. Operations related to truss topology (adding and removing joints and members) are connected by the thickest arrows, indicating a high probability that these operations will be employed together during truss design. Conversely, nontopology operations (changing the size of members, or moving joints) are connected by relatively thin arrows, indicating that these operations are far less likely to be applied together. However, these operations all have fairly high self-transition probabilities, meaning that they are likely to be applied several times in a row.

The higher-order Markov models are capable of explicitly representing long sequences of operations. On the other hand, the first-order Markov models assume that the selection of a subsequent operation is dependent on only the last operation, so that each operation is probabilistically linked to the next. Therefore, only pairs of operations can be represented explicitly. However, operation sequences of arbitrary length can be created by stringing together several of these probabilistically linked pairs. Graphically, the process of constructing these sequences consists of tracing a path through the graph-based representation of the transition matrix shown in Fig. 5. A set of high likelihood exemplar sequences produced through this process are provided in Fig. 6. Operations are shown in rectangular boxes, and the probability that the following operation would occur is given with a percentage over the linking arrow. The percentage of participants from the cognitive study who employed the sequence is also noted.

These sequences are multioperation 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 (and was employed by every participant in the cognitive study). Sequence B is similar to sequence A 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

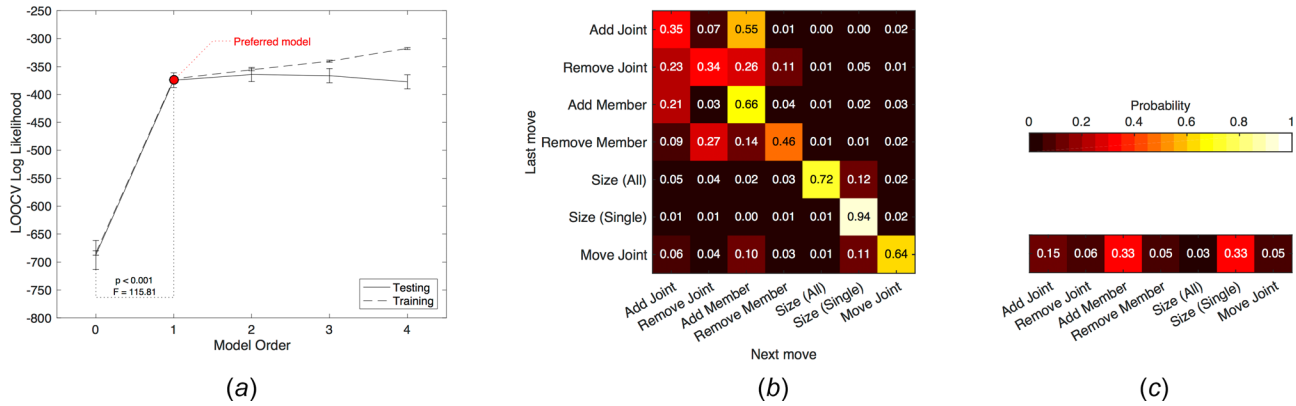


Fig. 4 Summary of results for truss design study, including (a) log-likelihood computed for models of increasing accuracy, (b) transition matrix of first-order Markov model, and (c) state frequencies for zero-order Markov model

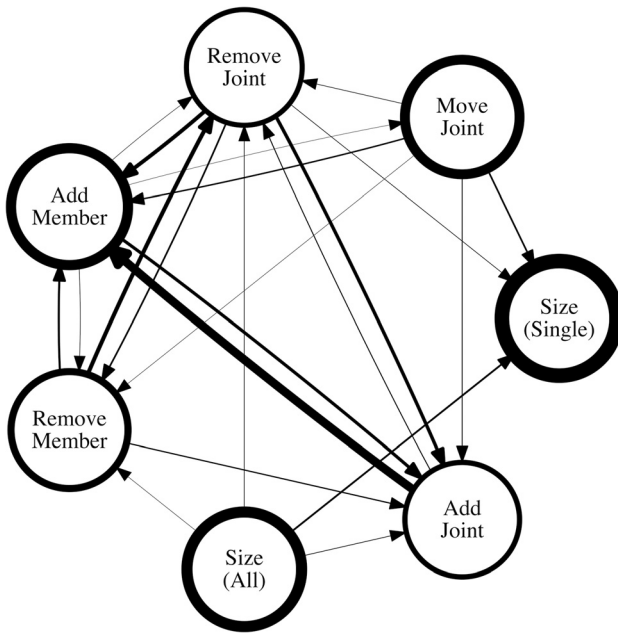


Fig. 5 Thresholded graph-based visualization of the first-order Markov model for the truss design study. Transition probabilities are depicted by the line thickness (or circle edge thickness for self-transitions).

section of the truss—removing a section of the truss with poor performance and then rebuilding it in an attempt to improve performance characteristics. Sequence B was employed by 92% of the cognitive study participants. 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.

4.3 Results for Home Cooling System Design Study. The same methodology used to analyze the truss design data was applied to operation data from the cooling system design study. A plot of the log-likelihood for models of increasing order is provided in Fig. 7(a). Many of the same trends from Fig. 4(a) 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. There is a minuscule mean increase in testing accuracy between the first-order and second-order models, but this increase is nonsignificant ($F = 0.02$, $p > 0.5$). After second-order, the training log-likelihood decreases, again indicating that the higher-order models tend to overfit the training data, losing generalizability. The marked divergence between training and testing curves displayed in Fig. 7(a) (which was not as sharp in Fig. 4) is indicative of the fact that the higher-order models have a greater tendency to overfit this dataset. This can be attributed to the fact that participants in the cooling system design study used far fewer operations than participants in the truss design study.

As with the analysis of the truss design study, the first-order Markov model is the preferred model for this design task. This may indicate that the sequencing of design operations can be treated as a first-order Markov process for the types of configuration tasks examined in this work, or perhaps more generally. At the very least, it is evidence that lower-order processes (but not zero-order) tend to be the most veridical. The higher-order models

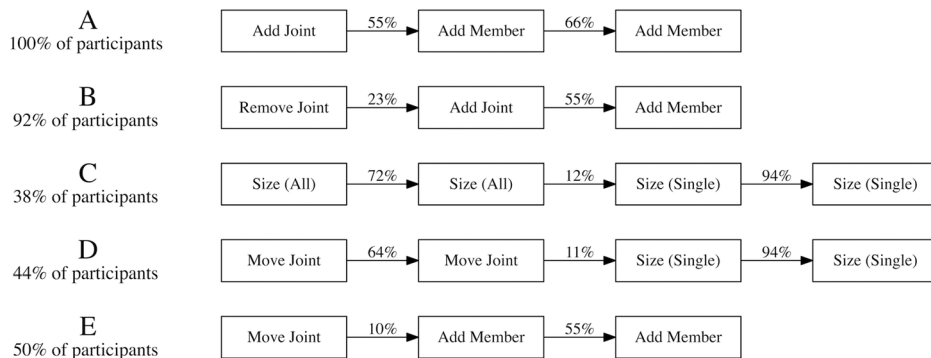


Fig. 6 Exemplar sequences extracted from the first-order Markov model for truss design study

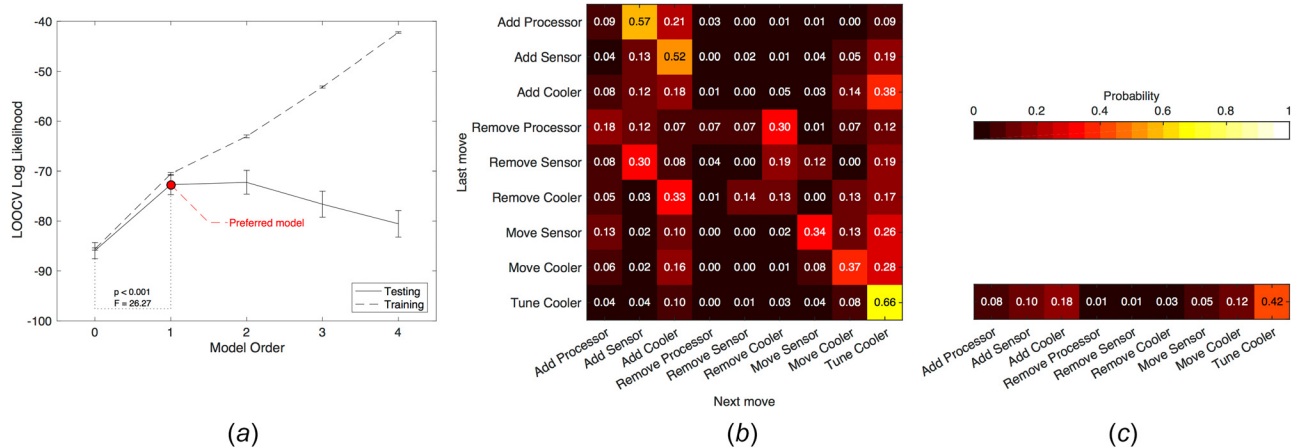


Fig. 7 Summary of results for cooling system design study, including (a) log-likelihood computed for models of increasing accuracy, (b) transition matrix of first-order Markov model, and (c) state frequencies for zero-order Markov model

may learn specific sequential constructs that are informative, but they do not appear to offer a superior description of aggregate patterns of design activity.

Examining the differences between Figs. 7(b) and 7(c) can once again provide insight as to the benefit that is derived from pursuing operations sequentially for the cooling system design task. Whereas the first-order Markov model developed for the truss design data had a strongly diagonal structure, the transition matrix developed for the cooling system data has several 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 Fig. 8 (thresholded in the same manner as Fig. 5).

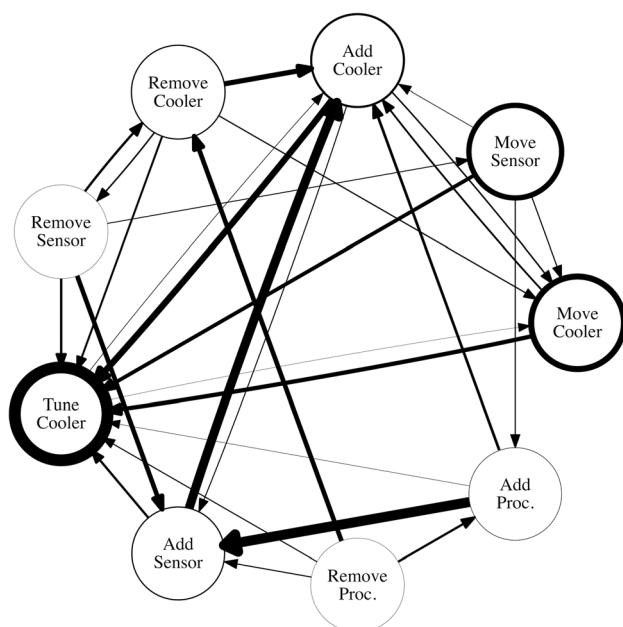


Fig. 8 Thresholded graph-based visualization of the first-order Markov model for the cooling system design study. Transition probabilities are depicted by the line thickness (or circle edge thickness for self-transitions).

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.

As shown in the results from the truss design study, longer sequences of operations can be extracted by traversing the graph-based representation of the transition matrix (see Fig. 9). Sequence A consists of a processor addition, a sensor addition, and a cooler addition. As noted above in Fig. 8, this sequence encodes the construction of the simplest independent subsystem possible, consisting of a sensor, a cooler, and a processor. Sequence B describes the removal followed by the addition of a cooler. These actions were necessary to transfer the control of a cooler to a different processor—this was not enabled with a single move during the study. Interestingly, the probability of the opposite of these two actions (adding a cooler and then deleting a cooler) was nearly 0. Sequences C, D, and E 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 Fig. 7(b)). Sequence C describes the common action sequence of adding a cooler and then immediately tuning its properties (a sequence employed by nearly every participant). An alternative cooler-related sequence is presented with sequence D 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 E is related to sequence D in that it consists of the same operations but they are enacted in a different order. Sequence E begins with moving a cooler, an action that might leave part of the building undercooled. A new cooler is then added (ostensibly in the undercooled area) and tuned to optimize performance.

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 in a sequence was based on knowledge of 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 the first-order Markov chains; the 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. The zero-order Markov models cannot encode sequence information, while the 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, the first-order Markov models fit the operation data better than the 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. The 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 Figs. 6 and 9, the longer sequences extracted by this method describe meaningful patterns of design that were employed often by study participants. These longer sequences might be represented more explicitly in the higher-order Markov chains, but they are represented both succinctly and accurately using the first-order Markov chains.

The number of independent parameters required to fully define the transition matrix for a Markov chain model is $(k - 1)k^m$, where k is the number of possible operations and m is the order of the model. This means that the number of model parameters increases rapidly (exponentially) with the order of the model. As an illustration, consider the truss design problem which has seven operations; a zero-order Markov chain requires the estimation of six independent parameters, a first-order model requires 42, and a second-order model requires 294 parameters. The fourth-order model trained in this work required the estimation of more than 10,000 independent parameters. Larger numbers of parameters require larger quantities of training data in order to accurately estimate the values of the parameters. This may offer some intuition as to why the first-order models in this work were the most veridical in comparison to human data. The higher-order sequences

simply require too much information and are thus too burdensome to learn. This could heavily bias human problem-solvers toward the lower-order sequences that can be learned with exponentially less information, enabling quick adaptation to new problems.

5 Investigation 2: Benefit of Operation Sequences

The first investigation discovered that the operation sequences employed by the study participants are accurately represented using the first-order Markov chains. However, it has not been shown directly that sequence learning positively impacts solution quality. On the one hand, the ability to learn sequences may help designers learn and exploit problem-specific heuristics to quickly find fruitful regions of the design space. On the other hand, sequence learning may bias designers toward learning sequences that greedily improve solution quality. Applying these greedy sequences could critically limit the breadth of search and lead designers toward local minima of inferior solution quality.

Because humans are capable of learning sequences implicitly [12,13], it is difficult to control and observe sequence learning as an experimental variable in a study with human participants. It is possible that implicit learning processes could take over even if participants were somehow prohibited from engaging in explicit sequence learning. For that reason, this work utilizes the CISAT modeling framework [2] to test the effects of the first-order sequence learning. The objects simulated in CISAT (i.e., designers and design teams) have explicitly defined protocols and skills. This makes it possible to directly modulate the degree to which CISAT agents engage in sequence learning, which in turn enables a direct comparison between sequential and nonsequential learning patterns. This assessment has the potential to indicate the degree to which sequence learning is or is not beneficial for real human designers.

5.1 Methodology. The CISAT modeling framework is an agent-based computational platform that is intended to simulate the process and performance of engineering design teams, and has been shown to do so accurately on the configuration-style design problems used in the current work [2]. The core functionality of the CISAT framework is based on a simulated annealing algorithm. This core functionality is augmented with eight cognitive characteristics, selected from the literature on design and problem-solving, in order to support a more veridical representation of the way in which individuals search for solutions and interact with a team while doing so [2]. It should be noted that the ability to learn and employ sequences is a characteristic of individuals. However, the corpus of data used here (from both the truss and cooling system design tasks) was produced by individual designers operating within teams. Therefore, the CISAT simulations in this work are structured to simulate the performance of teams. The sequence learning ability is implemented in CISAT at

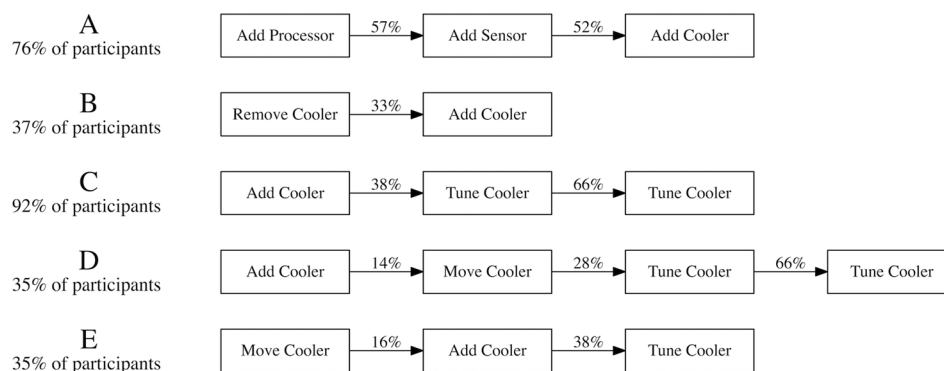


Fig. 9 Exemplar sequences extracted from the first-order Markov model for cooling system design study

the agent level, reflecting the individual sequence-learning abilities of human designers.

CISAT agents learn how to apply operations through operational learning, one of the eight cognitive characteristics implemented in CISAT. In Ref. [2], the operational learning characteristic was implemented through the following steps. At the beginning of each 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, \mathbf{p} . The chosen move operator, i , is then applied to 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 according to the update rule

$$p_i \leftarrow p_i \cdot (1 + k_{OL}) \quad (3)$$

where k_{OL} is a parameter that modifies how rapidly the values of \mathbf{p} are updated. If the move operator decreases the quality of the solution, the probability of selecting it in the future is reduced according to a similar update rule

$$p_i \leftarrow p_i \cdot (1 - k_{OL}) \quad (4)$$

The probability vector is renormalized following every update. Updating selection probability based on the effect of only the most recent application of a given move operator (instead of some average over past applications) reflects availability bias [51], the tendency of humans to place greater weight on information that is readily available in memory.

A second version of the operational learning characteristic was implemented for this work using the first-order Markov chain constructs. This model was chosen specifically because it was shown in Sec. 4 to accurately encode human operation sequences. This concept is implemented so that agents first select which operator to apply by randomly drawing from the probabilities defined in \mathbf{T} . Next, the agents update the matrix element corresponding to the operator that they chose, iteratively constructing a transition matrix that encodes the most beneficial move operator sequences. This two-step procedure is similar to the procedure identified in humans solving the Thurstone letter series completion task. Here, the updating of probabilities in \mathbf{T} aligns with the pattern generation step for the Thurstone task, and the selection of a specific operation based on the probabilities in \mathbf{T} aligns with the sequence generation step in the Thurstone task.

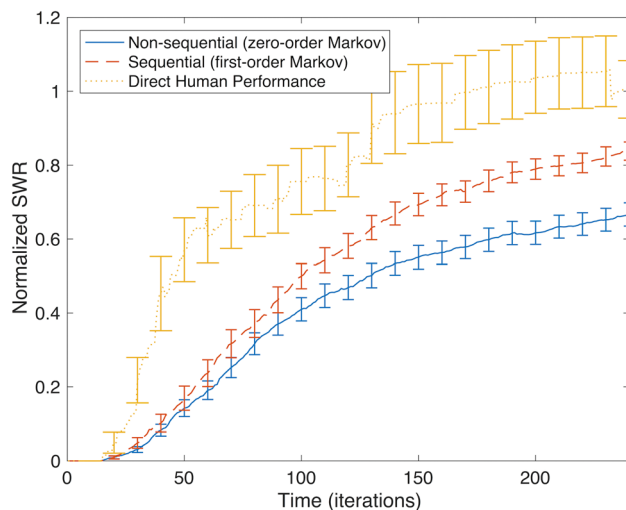


Fig. 10 Comparison of truss design quality for zero-order Markov model, first-order Markov model, and human performance on the truss design problem (error bars show ± 1 SE)

More specifically, when selecting a move operator, a random draw is taken from the multinomial distribution defined by row i of matrix \mathbf{T} , where operator i is the last move operator that was applied. The operator chosen, operator j , is then applied to the current solution. The probabilities contained in the transition matrix are then updated depending on whether operator j improves the quality of the current solution

$$T_{ij} \leftarrow T_{ij} \cdot (1 + k_{OL}) \quad (5)$$

or worsens it

$$T_{ij} \leftarrow T_{ij} \cdot (1 - k_{OL}) \quad (6)$$

This selection process probabilistically links the choice of the next move operator to the last move operator that was chosen via the Markov chain transition matrix, making it possible for CISAT agents to recognize and use beneficial sequences of operations. It should be noted that the transition matrix of a first-order Markov chain does not explicitly encode finite sequences of operations. Instead, the sequences are encoded probabilistically and implicitly based on the effects of applying move operators, which increases the likelihood of applying beneficial sequences. This probabilistic scheme also allows for new sequences to be discovered.

Only the operational learning characteristic was modified—all other agent protocols and characteristics are as originally given in Ref. [2]. This change is not intended to be an optimal learning approach, but rather to reflect the actuality of human behavior. If principles of the first-order sequencing are repurposed for use in computational synthesis algorithms, it may be more useful to update operator selection probabilities based on a measure of average performance rather than using the binary tuning approach featured here.

5.2 Results for Truss Design Study. In the interest of simplicity, performance was only simulated for the initial problem statement from the truss design study. A total of 100 teams were simulated for each of the two conditions: sequential, utilizing the first-order Markov chain concepts; and nonsequential, utilizing the zero-order Markov chains. Simulation results were then analyzed to extract each team's best solution at every iteration. A comparison of the two conditions is provided in Fig. 10, showing the mean normalized strength-to-weight ratio as a measure of solution quality.

The difference in final design quality between the two simulated conditions (zero-order Markov and first-order Markov) is highly significant ($F = 11.2$, $p < 0.001$), and the condition using the first-order Markov chain learning approach achieved a higher final design quality. Although the introduction of sequence-learning abilities does not raise CISAT solution quality to the level of the real human teams, it closes the difference between the two by nearly half, indicating that the ability to learn sequences is vital to the success of real designers.

5.3 Results for Home Cooling System Design Study.

CISAT was also used to simulate the performance of human design teams on the cooling system design task. Sequential and nonsequential learning were implemented as above with the first-order and zero-order Markov chains, respectively. The results of the simulations were then postprocessed to track each team's best solution over time. In this case, the 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 Fig. 11.

For this task, simulated teams that were capable of learning and employing sequences of operations achieved solutions with

significantly higher quality ($F = 5.91, p < 0.05$). The increase in solution quality that results from the introduction of sequence-learning abilities again helps to close the gap between simulation and real human performance.

5.4 Discussion. The objective of this section was to assess whether or not the ability to learn sequences contributes positively to eventual solution quality. This was accomplished by modifying the operational learning characteristic of CISAT to enable agents to learn beneficial sequences of operations by reinforcing a first-order Markov transition matrix. This made it possible to perform a comparison between CISAT-simulated teams employing either nonsequential learning (a zero-order Markov model) or sequential learning similar to that observed in designers (a first-order Markov model). Simulations were conducted to reflect cognitive studies involving the design of trusses and the design of cooling systems, and in both cases, sequential learning produced solutions with higher quality. This indicates that sequence learning is a beneficial aspect of human cognition during design.

The performance of sequence-learning and nonsequence-learning approaches is similar for the initial portion of the simulation on both design problems. A similar phenomenon was observed in other work that used machine learning to recognize and employ move operation pairs during computational design [38]. In that work, algorithms with and without the ability to learn pairs were compared and showed nearly identical performance for the first 3% of the search. In that work, as well as the current paper, the identical early performance can be explained as an exploratory phase—the agent or algorithm is still learning about the design space and has not yet learned effective move pairs or operation sequences. Once sufficient exploration has occurred, the agent or algorithm can begin to employ learned patterns to more effectively create solutions. This highlights the fact that, especially for sequence learning, exploration and exploitation are inextricably linked. Human designers may stand to benefit from emphasizing the recognition of beneficial operation sequences during early exploration in order to aid more effective exploitation during the later stages of design.

Although the addition of first-order sequence learning boosts performance when compared to nonsequence-learning simulations, there remains a substantial division between the quality of solutions produced in the simulations and those produced by humans. While the order in which operations are performed is important for exploration and exploitation of the design space, the way in which the operations are applied to the current solution

(e.g., which structural member is increased in size, or where a new cooler is added) is important as well. Since CISAT agents stochastically apply operations once chosen, it is likely that the gap between the performance of sequence-learning agents and humans is due to nuance in the application.

6 Conclusions

This paper investigated the sequencing of operations in engineering design problems using a variety of statistical and computational tools. Through analysis of human data from two cognitive studies, two research questions were specifically addressed:

- (1) *How much representational complexity is necessary to quantify the sequential patterns that designers employ during solving?* Markov chain models with increasing order (representative of how much memory is assumed in the model) were fit to the data from two human studies. For both studies, the analysis indicated that the sequencing of design actions might be treated accurately as a first-order Markov process. It should be noted that longer finite-length sequences of operations may still be extracted by traversing the graphs described by the first-order Markov transition matrices.
- (2) *Does the use of operation sequences benefit designers?* The CISAT modeling framework was used to assess the potential benefit from learning and employing operation sequences. Several sets of simulations were conducted in which teams of agents solved the design problems from the two cognitive studies, either with the ability to learn sequences (encoded within a first-order Markov model) or without that ability (represented mathematically by a nonsequential statistical model). A comparison of these simulations demonstrated that sequence-learning abilities significantly increased solution quality for both design problems.

The results of this work have the potential to inform novel approaches for design education and training. Here, it was shown that designers utilized first-order sequences of operations, and that this allowed them to discover solutions with higher quality. Although it is not clear whether these sequences were learned implicitly or explicitly, there is evidence that explicit awareness of a sequence-learning task can improve performance [24]. Therefore, it is possible that teaching designers to be aware of the importance of learning sequences could improve their ability to learn said sequences. This self-awareness could be augmented in computer-aided design software by logging and analyzing design activity to provide real-time feedback about common sequential strategies, ensuring explicit awareness. Examining and quantifying the difference between explicit and implicit sequence-learning modalities with applications to design will be addressed in future work.

This work also holds implications for design automation and synthesis. Specifically, this work demonstrates that it may be possible to use Markov constructs to improve the effectiveness of design algorithms. The second investigation of this work implemented sequence-learning abilities in CISAT, a computational framework that is based in part on principles of stochastic optimization. This enabled computational agents to create and modify solutions using sequential chains of operations, resulting in improved performance. A similar implementation could be used to imbue other design synthesis algorithms with the ability to learn and apply sequences of operations. Such applications hinge on the fact that Markov chain models are generative [52], meaning that they encode the training data in such a way that they can be used to create new, synthetic data. In a design context, this amounts to the creation of new operational sequences that adhere probabilistically to observed patterns. Markov chain models could be learned and reinforced as an algorithm creates design solutions, or trained prior to use in an algorithm if sufficient prior data are available.

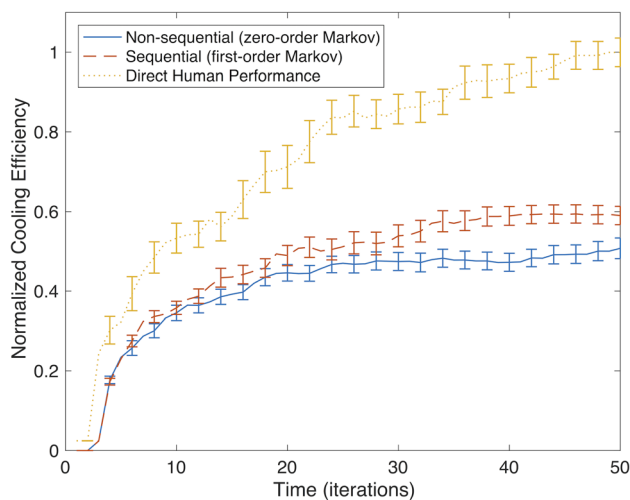


Fig. 11 Comparison of cooling system design quality for zero-order Markov model, first-order Markov model, and human performance on the cooling system design problem (error bars show ± 1 SE)

This work offers a foundation for describing sequence learning in engineering design by demonstrating that the first-order Markov chains are a veridical model, and that learning simple first-order sequences can improve the solution quality. These descriptive results establish a basis for future normative and explanatory research. Future explanatory work should investigate the underlying causation that gives rise to the sequential behaviors observed and described here, perhaps by using Markov decision process models [53]. Leveraging additional results from psychology (such as the importance of pauses during solving [54]) could also provide explanatory power. Further normative work could utilize numerical simulations to identify the dependence of learned sequences on the complexity and characteristics of the problem at hand by employing a predictive response surface methodology [47,55].

Acknowledgment

This material is based upon the work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE125252 and the United States Air Force Office of Scientific Research through Grant Nos. FA9550-12-1-0374 and FA9550-16-1-0049. 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. An early version of part of this work was included in the proceedings of the Conference on Design Computing and Cognition [56].

Funding Data

- Air Force Office of Scientific Research (Grant Nos. FA9550-12-1-0374 and FA9550-16-1-0049).
- Division of Graduate Education (Grant No. DGE125252).

References

- [1] 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. Studies*, **36**(1), pp. 99–121.
- [2] McComb, C., Cagan, J., and Kotovsky, K., 2015, "Lifting the Veil: Drawing Insights About Design Teams From a Cognitively-Inspired Computational Model," *Des. Studies*, **40**, pp. 119–142.
- [3] Neill, T. M., Gero, J. S., and Warren, J., 1998, "Understanding Conceptual Electronic Design Using Protocol Analysis," *Res. Eng. Des.*, **10**(3), pp. 129–140.
- [4] Kan, J. W., and Gero, J. S., 2011, "Comparing Designing Across Different Domains: An Exploratory Case Study," 18th International Conference on Engineering Design (ICED), Lyngby/Copenhagen, Denmark, Aug. 15–19, pp. 194–203.
- [5] Yu, R., and Gero, J. S., 2015, "An Empirical Foundation for Design Patterns in Parametric Design," 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRRIA), Daegu, South Korea, May 20–23, pp. 1–9.
- [6] Gero, J. S., and Peng, W., 2009, "Understanding Behaviors of a Constructive Memory Agent: A Markov Chain Analysis," *Knowl. Based Syst.*, **22**(8), pp. 610–621.
- [7] March, J. G., 1991, "Exploration and Exploitation in Organizational Learning," *Organ. Sci.*, **2**(1), pp. 71–87.
- [8] Gupta, A. K., Smith, K. G., and Shalley, C. E., 2006, "The Interplay Between Exploration and Exploitation," *Acad. Manage. J.*, **49**(4), pp. 693–706.
- [9] Goncher, A., Johri, A., Kothaneth, S., and Lohani, V., 2009, "Exploration and Exploitation in Engineering Design: Examining the Effects of Prior Knowledge on Creativity and Ideation," 39th IEEE Frontiers in Education Conference (FIE), San Antonio, TX, Oct. 18–21, pp. 1–7.
- [10] 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.
- [11] Kotovsky, K., and Simon, H. A., 1990, "What Makes Some Problems Really Hard: Explorations in the Problem Space of Difficulty," *Cognit. Psychol.*, **22**(2), pp. 143–183.
- [12] Nissen, M. J., and Bullemer, P., 1987, "Attentional Requirements of Learning: Evidence From Performance Measures," *Cognit. Psychol.*, **19**(1), pp. 1–32.
- [13] Reed, J., and Johnson, P., 1994, "Assessing Implicit Learning With Indirect Tests: Determining What Is Learned About Sequence Structure," *J. Exp. Psychol.*, **20**(3), pp. 585–594.
- [14] Clegg, B. A., Digriolamo, G. J., and Keele, S. W., 1998, "Sequence Learning," *Trends Cognit. Sci.*, **2**(8), pp. 275–281.
- [15] Chase, W. G., and Simon, H. A., 1973, "Perception in Chess," *Cognit. Psychol.*, **4**(1), pp. 55–81.
- [16] Egan, D. E., and Schwartz, B. J., 1979, "Chunking in Recall of Symbolic Drawings," *Mem. Cognit.*, **7**(2), pp. 149–158.

- [17] Reitman, J. S., 1976, "Skilled Perception in Go: Deducing Memory Structures From Inter-Response Times," *Cognit. Psychol.*, **8**(3), pp. 336–356.
- [18] Moss, J., Cagan, J., and Kotovsky, K., 2004, "Learning From Design Experience in an Agent-Based Design System," *Res. Eng. Des.*, **15**(2), pp. 77–92.
- [19] 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, New York.
- [20] Simon, H. A., and Kotovsky, K., 1963, "Human Acquisition of Concepts for Sequential Patterns," *Psychol. Rev.*, **70**(6), pp. 534–546.
- [21] Kotovsky, K., and Simon, H. A., 1973, "Empirical Tests of a Theory of Human Acquisition of Concepts for Sequential Patterns," *Cognit. Psychol.*, **4**(3), pp. 399–424.
- [22] Novick, L. R., and Tversky, B., 1987, "Cognitive Constraints on Ordering Operations: The Case of Geometric Analogies," *J. Exp. Psychol.: Gen.*, **116**(1), pp. 50–67.
- [23] Perruchet, P., and Amorim, M.-A., 1992, "Conscious Knowledge and Changes in Performance in Sequence Learning: Evidence Against Dissociation," *J. Exp. Psychol.*, **18**(4), pp. 785–800.
- [24] Willingham, D. B., Nissen, M. J., and Bullemer, P., 1989, "On the Development of Procedural Knowledge," *J. Exp. Psychol.*, **15**(6), pp. 1047–1060.
- [25] Curran, T., and Keele, S. W., 1993, "Attentional and Nonattentional Forms of Sequence Learning," *J. Exp. Psychol.*, **19**(1), pp. 189–202.
- [26] Stempfle, J., and Badke-schaub, P., 2002, "Thinking in Design Teams—An Analysis of Team Communication," *Des. Studies*, **23**(5), pp. 473–496.
- [27] 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.
- [28] Goldschmidt, G., and Rodgers, P. A., 2013, "The Design Thinking Approaches of Three Different Groups of Designers Based on Self-Reports," *Des. Studies*, **34**(4), pp. 454–471.
- [29] Radcliffe, D. F., and Lee, T. Y., 1989, "Design Methods Used by Undergraduate Engineering Students," *Des. Studies*, **10**(4), pp. 199–207.
- [30] Todd, D., 1997, "Multiple Criteria Genetic Algorithms in Engineering Design and Operation," Ph.D. thesis, University of Newcastle, Newcastle-upon-Tyne, UK.
- [31] Rogers, J., 1996, "DeMAID/GA—An Enhanced Design Manager's Aid for Intelligent Decomposition," *AIAA Paper No. 96-4157*.
- [32] Sen, C., Ameri, F., and Summers, J. D., 2010, "An Entropic Method for Sequencing Discrete Design Decisions," *ASME J. Mech. Des.*, **132**(10), p. 101004.
- [33] Waldron, M. B., and Waldron, K. J., 1988, "A Time Sequence Study of a Complex Mechanical System Design," *Des. Studies*, **9**(2), pp. 95–106.
- [34] Meier, C., Yassine, A. A., and Browning, T. R., 2007, "Design Process Sequencing With Competent Genetic Algorithms," *ASME J. Mech. Des.*, **129**(6), pp. 566–585.
- [35] Gero, J. S., 1990, "Design Prototypes: A Knowledge Representation Schema for Design," *AI Mag.*, **11**(4), pp. 26–36.
- [36] Bhatta, S. R., and Goel, A. K., 1992, "Discovery of Physical Principles From Design Experiences," *International Machine Learning Workshop*, San Mateo, CA [Int. J. AI EDAM **8**(2), pp. 1–22 (1994)].
- [37] Gero, J. S., Kan, J. W., and Pourmohamadi, M., 2011, "Analysing Design Protocols: Development of Methods and Tools," *Research Into Design*, Research Publishing, Singapore, pp. 3–10.
- [38] Vale, C. A. W., and Shea, K., 2003, "A Machine Learning-Based Approach to Accelerating Computational Design Synthesis," 14th International Conference on Engineering Design (ICED), Stockholm, Sweden, Aug. 19–21, pp. 183–184.
- [39] Stroock, D. W., 2005, *An Introduction to Markov Processes*, Springer Science and Business Media, Berlin.
- [40] Markov, A. A., 1907, "Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain," *The Notes of the Imperial Academy of Sciences of St. Petersburg*, VIII Series, Physio-Mathematical College XXII, Imperial Academy of Science, St. Petersburg, Russia.
- [41] Scherr, A. L., 1962, *An Analysis of Time-Shared Computer Systems*, Massachusetts Institute of Technology, Cambridge, MA.
- [42] Page, L., Brin, S., Motwani, R., and Winograd, T., 1999, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab, Stanford, CA, Technical Report No. 422.
- [43] Tamir, A., 1998, *Applications of Markov Chains in Chemical Engineering*, Elsevier, Amsterdam, The Netherlands.
- [44] Gero, J. S., and Kan, J. W., 2009, "Learning to Collaborate During Team Designing: Some Preliminary Results From Measurement-Based Tools," Third International Conference on Research Into Design Engineering (ICORD), Bangalore, India, Jan. 10–12, pp. 560–567.
- [45] Gero, J. S., and Kan, J. W., 2011, "Learning to Collaborate During Team Designing: Quantitative Measurement," Third International Conference on Research Into Design Engineering (ICORD), Bangalore, India, Jan. 10–12, pp. 978–981.
- [46] Raftery, A. E., 1985, "A Model for High-Order Markov Chains," *J. R. Stat. Soc. Ser. B*, **47**(3), pp. 528–539.
- [47] 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.
- [48] Arlot, S., and Celisse, A., 2010, "A Survey of Cross-Validation Procedures for Model Selection," *Stat. Surv.*, **4**, pp. 40–79.
- [49] Stone, M., 1974, "Cross-Validatory Choice and Assessment of Statistical Predictions," *J. R. Stat. Soc. Ser. B*, **36**(2), pp. 111–147.

- [50] Hastie, T., Tibshirani, R., and Friedman, J., 2009, *The Elements of Statistical Learning*, Springer, New York.
- [51] Tversky, A., and Kahneman, D., 1973, "Availability: A Heuristic for Judging Frequency and Probability," *Cognit. Psychol.*, **5**(2), pp. 207–232.
- [52] 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., Oxford University Press, Oxford, UK, pp. 3–24.
- [53] Bellman, R., 1957, "A Markovian Decision Process," *J. Math. Mech.*, **6**(5), pp. 679–684.
- [54] Chi, M. T. H., Glaser, R., and Rees, E., 1982, "Expertise in Problem Solving," *Advances in the Psychology of Human Intelligence*, Erlbaum, Hillsdale, NJ, pp. 7–75.
- [55] Dinar, M., Park, Y.-S., Shah, J. J., and Langley, P., 2015, "Patterns of Creative Design: Predicting Ideation From Problem Formulation," *ASME Paper No. DETC2015-46537*.
- [56] McComb, C., Cagan, J., and Kotovsky, K., 2016, "Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks," *Design Computing and Cognition '16*, J. S. Gero, ed., Springer International Publishing, Cham, Switzerland, pp. 401–418.