

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 (Mittal and Frayman, 1989). 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. Such descriptive research is a necessary first step before any attempt at a prescriptive methodology can be undertaken. Thus, the overarching goals of this work are (1) to analyze the results of a cognitive study of configuration design to provide a descriptive model, and (2) to extract beneficial prescriptive heuristics by comparing performance-differentiated models. The approach used to extract these heuristics is formal and novel. We mine the cognitive study data using hidden Markov models to automatically create representations of design behavior and process. It is shown that hidden Markov models trained on the data are sensitive enough to discover and clarify the procedural differences between high- and low-performing designers. A comparison of these performance-differentiated models provides insights that constitute heuristics for effective configuration design.

This work specifically mines data generated by human participants during a truss design task. In that study, the solving approach used by participants generally conformed to a propose-critique-modify methodology, one frequently identified in configuration design (Chandrasekaran, 1990). 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 (Wielinga and Schreiber, 1997) provide a structured approach to searching the solution space associated with a given configuration problem. However, the solution space for configuration design problems branches polynomially (Mittal and Frayman, 1989), 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. Through a data-mining approach this work will extract and codify any such 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). In a cognitive 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 (McComb et al., 2015a). 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 (McComb et al., 2015a). 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 (McComb et al., 2016). It was revealed that sequencing was important for achieving high quality designs. Specifically, topology operations were often applied together 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 (Kim et al., 2008).

In the later stages of the design process, designers are known to frequently switch between embodiment and detail design activities (Snider et al., 2014). Frequent switching between different levels of detail and/or abstraction can be an indication of opportunistic design behavior (Guindon, 1990; Visser, 1994). Bender and Blessing (2004) linked the concept of opportunism in design to the use of strongly prescriptive methodologies in early 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 an opportunistic approach to solving with a small degree of hierarchical structure (Bender and Blessing, 2004).

In order to direct designers towards effective configuration design procedures, it is necessary to identify heuristics that can be used to reduce the effective size of the solution space. The treatment of

designer activity as a hidden Markov process in this work makes it possible to automatically infer process models from a log of designer activity, from which heuristics can then be identified. The following sections will provide background on hidden Markov models, introduce the human study data set that is examined in this work, and detail the methodology used to mine heuristics from the data set. Next, hidden Markov models are trained on the entire data set in order to identify aggregate procedural characteristics, and separate hidden Markov models are also trained on high-performing and low-performing segments of the data. A comparison of these performance-differentiated models indicates that there may exist generalizable heuristics that improve human performance on configuration design problems. These heuristics can be provided to designers to increase the efficiency of design space search without choking out the potential for opportunistic processing.

2 HIDDEN MARKOV MODELS

A hidden Markov model describes a stochastic process in which a system transitions between a finite number of discrete states that cannot be directly observed (Durbin et al., 1998). 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, T , and the emission matrix, E . The transition matrix contains the probability of transitioning to a future hidden state from the current hidden 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 hidden 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 and colleagues (Baum, 1972; Baum et al., 1970; Baum and Eagon, 1967; Baum and Petrie, 1966; Baum and Sell, 1968). One of the first practical uses of hidden Markov models was for speech recognition (Jelinek et al., 1975), but they have also been utilized in fields as diverse as protein modeling (Krogh et al., 1994), economic forecasting (Gonzalez et al., 2005), team military tactics (White et al., 2009), and cognitive skill acquisition (Tenison and Anderson, 2016). 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 arrows, and the emissions are shown with dashed arrows. The entries of the emission matrix are shown next to the dashed arrows that indicate the relevant emission, and transition probabilities are shown next to the solid arrows denoting the relevant transition.

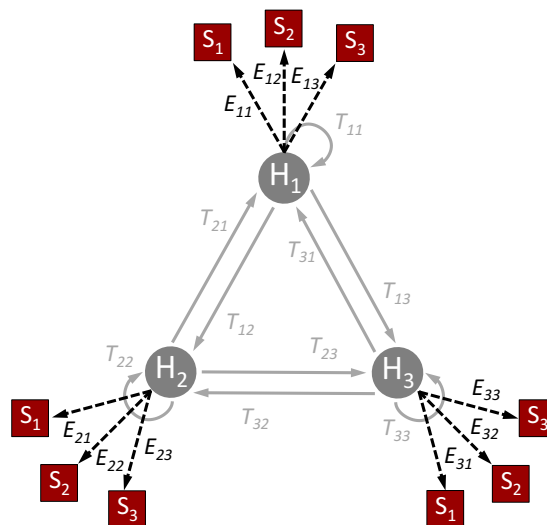


Figure 1. Example of a hidden Markov model with three states and three emission tokens.

In the current work, the tokens emitted by the hidden states are design operations. By treating the design operations as probabilistic representations, the hidden states that constitute the model become the underlying cognitive or procedural states that the designer goes through during the design process.

3 TRUSS DESIGN DATA SET

In a previously conducted experiment, teams of three senior mechanical engineering students were tasked with the design of a truss structure. 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.

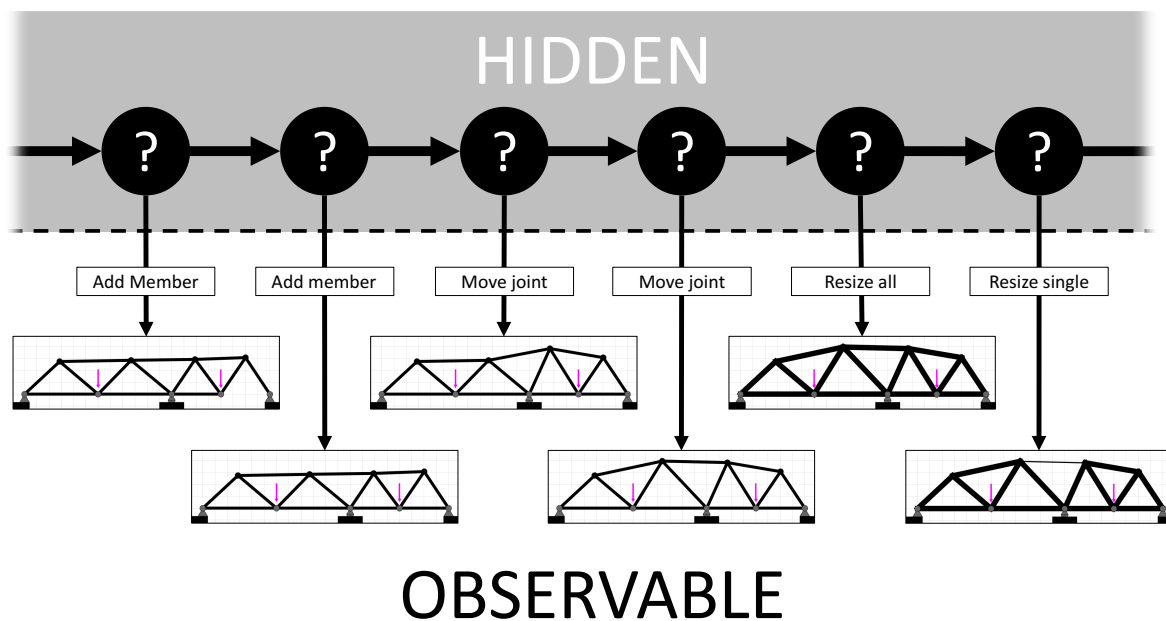


Figure 2. Example truss operation sequences.

Every participant was given access to a graphical truss design computer program to facilitate completion of the design task. Through this interface, participants could construct, analyze, and share trusses within their team. 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. Every participant performed an average of 400-500 such operations. A short example operation sequence is provided in Figure 2. The truss design problem constitutes a *full configuration design* problem, per the guidelines provided by Wielinga and Schreiber (1997). This type of problem is characterized by parametrized 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).

4 APPROACH AND ANALYSIS

The current work analyzes data from the study in the previous section using Hidden Markov models. This section outlines the specific methodology used in that analysis. The first sub-section outlines the Baum-Welch algorithm. This algorithm provides a means for computing transmission and emission matrices for a hidden Markov model based on an observed sequence of tokens. The Baum-Welch algorithm assumes that the number of hidden states is known, but such information is not known for

the current application. For that reason, the second sub-section outlines the procedure used here to estimate the correct number of hidden states. The final sub-section details the procedure used for identifying high- and low-performing designers.

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 (Baum et al., 1970). This algorithm uses an expectation-maximization approach (Dempster et al., 1977) 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 forwards-backwards algorithm to compute the probability that every observation in the training data resulted from any state in the model (Stratonovich, 1960). 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 (Durbin et al., 1998).

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 a number of models with varying values of k , and then compare them in some way. In this work we trained several models with values of k from one to a maximum of seven, the number of operations associated with the design problem. 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). For each value of k , models were trained using leave-one-out cross-validation (Arlot and Celisse, 2010). For a data set consisting of n samples, this cross-validation approach trains a model with $n - 1$ samples, and then tests the model on the sample that was not used for training. This procedure is repeated until every individual sample has been used for testing.

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

4.3 Comparing High- and Low-performing Designers

A key aspect of this work is the utilization of hidden Markov models to compare the procedures used by high-performing designers and low-performing designers. Of the 48 individuals who took part in the truss design study, 15 individuals (31.25% of the population) were designated as high-performing designers and 15 individuals (31.25% of the population) were designated 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 (McComb et al., 2015b).

5 RESULTS

Figure 3 shows the results of training hidden Markov models with varying numbers of hidden states on the data from all participants of the truss design study. 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, the four-state model is selected as the preferred model in this application.

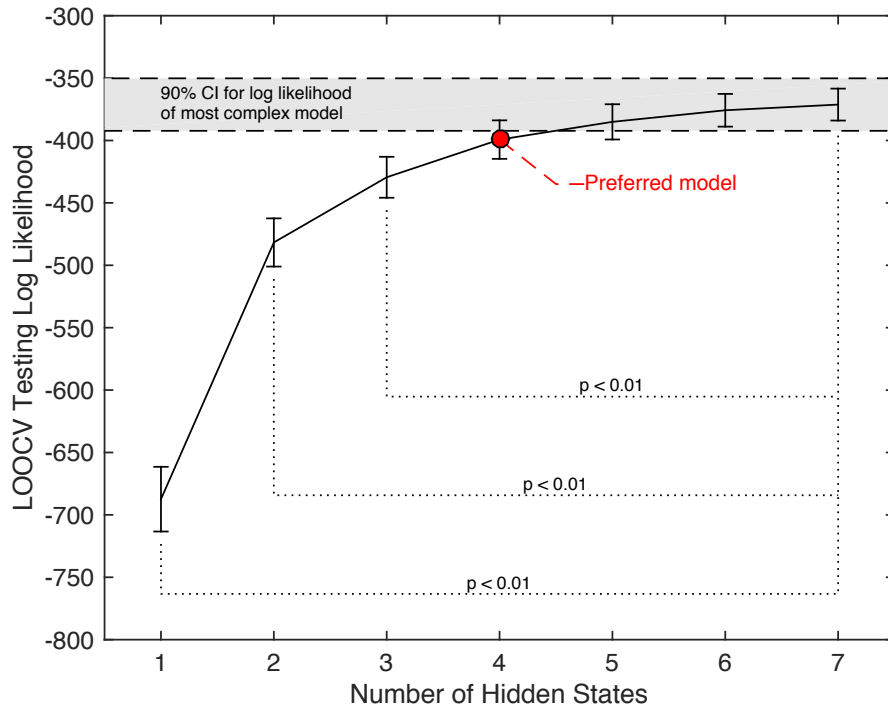


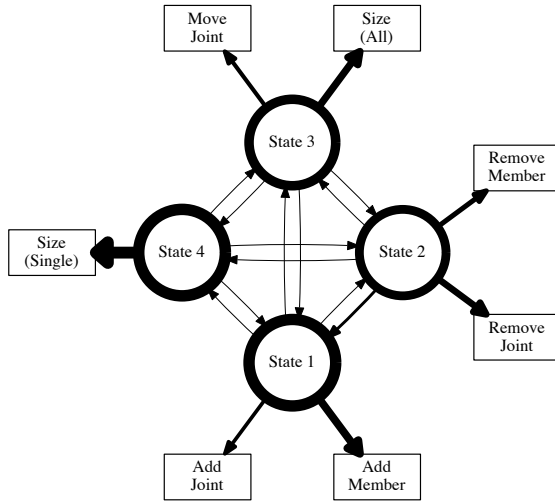
Figure 3. Testing log-likelihood on Truss Design Study data for models with increasing number of hidden states (error bars show ± 1 S.E.).

Hidden Markov models were also trained separately on the data from the high-performing and low-performing designers from the study, using four hidden states as chosen based on Figure 3. A graph-based representation of each of these four-state hidden Markov models is provided in Figure 4. Circular nodes indicate the hidden states, numbered 1 through 4. These states are numbered according to the ordering with the highest probability, with State 1 being the most likely initial state, and State 4 being the most likely terminal state. 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).

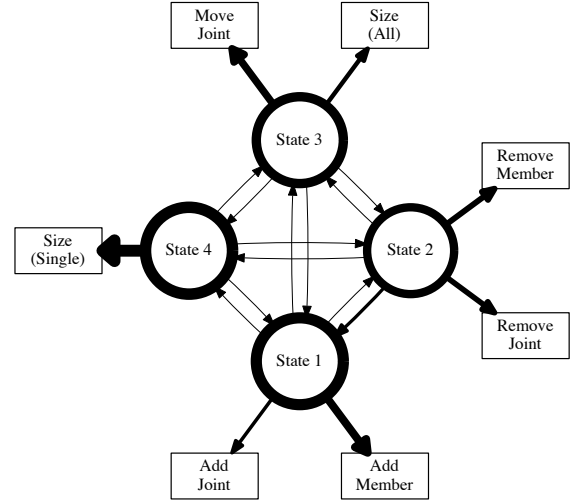
Interestingly, the model trained on the low-performing designers (see Figure 4(a)) is structurally identical to that trained on all designers (see Figure 4(b)), although specific probabilities differ slightly. This indicates that the general procedural characteristics encoded in these models were followed by the majority of study participants. In States 1 and 2 of these models, 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 operations) while State 2 corresponds to destruction of parts of the truss topology (through joint and member removal operations). There is a greater chance of transitioning from State 2 to State 1 than of the opposite transition (as indicated by the weights of the arrows between the two states). This indicates that the default topology mode of designers during the study was construction (State 1); transitions to the destructive state (State 2) were rare and did not last long before a return to construction. In State 3 the designer performs operations to reposition joints or for modification of the size of all structural members simultaneously. A designer in State 3 has a 78% chance of remaining in that state (indicated by the heavy weight of the border of State 3). 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 parameter state (8%). Once a designer reaches State 4 they have a very high probability of remaining in that state (94%, evidenced by the thick black border around the state). 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 nearing completion.

The model trained on data from the high-performing designers, shown in Figure 4(c), shows distinct differences from both the aggregate model and the low-performing model. The primary structural

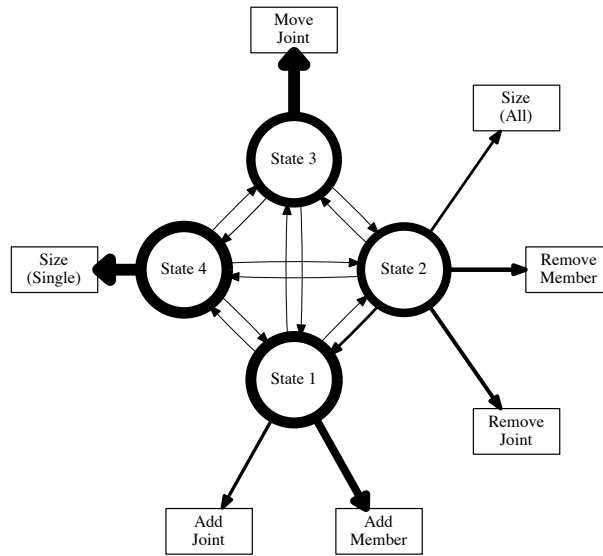
difference in the model is that the operation for global resizing of structural members (“Size (All)”) moves from State 3 to State 2. This indicates the introduction of a parameter operation to a state that was previously dominated by topology operations (namely, operations for removing elements from the truss), demonstrating 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 (similar to how State 4 indicates a sweep through the design to adjust member size).



(a) Model trained on data from low-performing designers.



(b) Model trained on data from all designers.



(c) Model trained on data from high-performing designers

Figure 4. Visual representation of four-state hidden Markov models.

6 DISCUSSION

A comparison of the hidden Markov models trained on data from high- and low-performing designers is provided in Table 1. The benefit derived from the differences between these models is likely

twofold. First, the devotion of State 3 to a spatial operation 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 – moving a joint can make it necessary to re-adjust the size of a member that has already been fine-tuned. 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 their early concepts. 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 granular improvements at an early stage. However, this activity is better described as opportunism, and aligns with other work that has demonstrated opportunistic activity in designer behavior and correlated that opportunism with beneficial design outcomes (Bender and Blessing, 2004; Guindon, 1990; Visser, 1994).

Table 1. Comparison of models for high- and low-performing designers.

State Number	Low-performing	High-performing
1	Topology – Add member Topology – Add joint	Topology – Add member Topology – Add joint
2	Topology – Remove member Topology – Remove joint —	Topology – Remove member Topology – Remove joint Parameter – Size (all)
3	Spatial – Move joint Parameter – Size (all)	Spatial – Move joint —
4	Parameter – Size (single)	Parameter – Size (Single)

This pattern of activity is also similar (at least analogically) to A* search (Hart et al., 1968). The A* algorithm is used for finding the minimum cost path between two points (typically represented as nodes in a graph) and follows a best-first search pattern. Search is specifically guided by an estimate of total path cost that is the sum of the cost of the path from the starting node to the current node (committed cost) and a heuristic estimate of the cost of the path from the current node to the goal node (expected cost). In this work, the current performance of a given solution topology is analogous to the committed cost, while the analogical equivalent of the expected cost is estimated differently by the high- and low-performing designers. Low-performing designers took the current performance as a direct indication of the expected cost heuristic, which resulted in relatively naïve early search. To maintain the search analogy, this could be likened to breadth-first search. In contrast, high-performing designers performed some parameter modifications to early topologies, which allowed them to more accurately estimate an expected cost heuristic. This provided better design performance information to guide early search towards a system topology with a high likelihood of delivering high final performance. To return to the search analogy, high-performing designers departed from the breadth-first style employed by low-performing designers and achieved a heuristic style that can be likened to the A* algorithm.

7 CONCLUSION

Configuration design is a common task in engineering, yet few studies have been conducted to examine how humans naturally engage in the task. Such descriptive research is necessary before prescriptive recommendations can be created. Thus, the objectives of this work were (1) to produce a descriptive model of aggregate human behavior in configuration design, and (2) to extract beneficial

prescriptive heuristics by comparing models trained on different segments of the design data. Hidden Markov models were utilized as a data-mining technique to fulfill both of these objectives.

This work first applied hidden Markov models to the entirety of the data from the cognitive study, producing an aggregate model of designer activity. The best model for the given data contained four hidden states. The states in the model aligned with specific activities: 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. It is unclear if the number and operation type of the hidden states is due to a general pattern of human cognition, is reflective specifically of the truss design task, or is due to some combination of influences both internal and external to the designer.

Next, separate hidden Markov models were trained on the data from high- and low-performing designers. The model trained on the low-performing designers was identical in structure to the model trained on all designers, but the model trained on the high-performing designers showed different patterns of activity. The key structural difference between the models was that high-performing designers incorporated parameter operations in an early state that were otherwise dominated by topology operations. 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 those early concepts, providing more accurate information to guide their search process.

The propensity of high-performing designers to use topology and parameter operations together in the early phases of their process is similar to opportunistic behavior that has been observed and correlated with high performance in other domains (Bender and Blessing, 2004; Guindon, 1990; Visser, 1994). This process also bears some similarity, at least analogically, to the A* search algorithm. That algorithm is used for path-finding and employs a heuristic estimate of the final path cost (Hart et al., 1968). That heuristic estimate is similar to the way in which high-performing designers were able to estimate the final quality of solutions by judiciously applying early parameter operations.

This work showed that hidden Markov models can be an effective tool for automating the construction of models for design processes. By employing this statistical tool, it was also 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. This approach constitutes a heuristic that can be provided to designers to increase the efficiency of design space search without choking out the potential for opportunistic processing. Future work should apply the methodology used here to other design data sets (both for configuration design and other tasks) to examine the extent to which the results can be generalized. The heuristic of coevolving topology and parameters (and variants of it) should also be experimentally validated as a means of demonstrably improving solution quality.

ACKNOWLEDGMENTS

This material is based upon 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 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.

REFERENCES

- Arlot, S. and Celisse, A. (2010), "A survey of cross-validation procedures for model selection", *Statistics Surveys*, Vol. 4, pp. 40–79.
- Baum, L.E. (1972), "An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes", *Inequalities*, Vol. 3, pp. 1–8.
- Baum, L.E. and Eagon, J.A. (1967), "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology", *Bulletin of the American Mathematical Society*, Vol. 73 No. 3, pp. 360–364.
- Baum, L.E. and Petrie, T. (1966), "Statistical Inference for Probabilistic Functions of Finite State Markov Chains", *The Annals of Mathematical Statistics*, Vol. 37 No. 6, pp. 1554–1563.
- Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970), "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains", *The Annals of Mathematical Statistics*, Vol. 41 No. 1, pp. 164–171.
- Baum, L.E. and Sell, G. (1968), "Growth transformations for functions on manifolds", *Pacific Journal of*

Mathematics, Vol. 27 No. 2, pp. 211–227.

- Bender, B. and Blessing, L. (2004), “On the superiority of opportunistic design strategies during early embodiment design”, *Proceedings of DESIGN 2004, the 8th International Design Conference*, pp. 1–6.
- Chandrasekaran, B. (1990), “Design Problem Solving: A Task Analysis”, *AI Magazine*, Vol. 11 No. 4, pp. 59–71.
- Dempster, A., Laird, N.M. and Rubin, D.B. (1977), “Maximum Likelihood from Incomplete Data via the EM Algorithm”, *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39 No. 1, pp. 1–38.
- Durbin, R., Eddy, S.R., Krogh, A. and Mitchison, G. (1998), *Biological Sequence Analysis*, Cambridge University Press, Cambridge.
- Gonzalez, A.M., Roque, A.M.S. and Garcia-Gonzalez, J. (2005), “Modeling and Forecasting Electricity Prices with Input/Output Hidden Markov Models”, *IEEE Transactions on Power Systems*, Vol. 20 No. 1, pp. 13–24.
- Guindon, R. (1990), “Designing the Design Process: Exploiting Opportunistic Thoughts”, *Human-Computer Interaction*, Vol. 5 No. 2, pp. 305–344.
- Hart, P., Nilsson, N. and Raphael, B. (1968), “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”, *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4 No. 2, pp. 100–107.
- Jelinek, F., Bahl, L. and Mercer, R. (1975), “Design of a linguistic statistical decoder for the recognition of continuous speech”, *IEEE Transactions on Information Theory*, Vol. 21 No. 3, pp. 250–256.
- Kim, M.J., Korea, S. and Maher, M. Lou. (2008), “The impact of tangible user interfaces on spatial cognition during collaborative design”, *Design Studies*, Vol. 29 No. 3, pp. 222–253.
- Krogh, A., Brown, M., Mian, I.S., Sjölander, K. and Haussler, D. (1994), “Hidden Markov models in computational biology. Applications to protein modeling.”, *Journal of Molecular Biology*, Vol. 235 No. 5, pp. 1501–1531.
- McComb, C., Cagan, J. and Kotovsky, K. (2015a), “Lifting the Veil: Drawing insights about design teams from a cognitively-inspired computational model”, *Design Studies*, Vol. 40, pp. 119–142.
- McComb, C., Cagan, J. and Kotovsky, K. (2015b), “Rolling with the punches: An examination of team performance in a design task subject to drastic changes”, *Design Studies*, Elsevier Ltd, Vol. 36 No. 1, pp. 99–121.
- McComb, C., Cagan, J. and Kotovsky, K. (2016), “Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks”, in Gero, J.S. (Ed.), *Design Computing and Cognition '16*, Springer International Publishing, Cham, Switzerland, pp. 401–418.
- Mittal, S. and Frayman, F. (1989), “Towards a Generic Model of Configuration Tasks”, *IJCAI'89 Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 2*, pp. 1395–1401.
- Snider, C., Dekoninck, E. a and Culley, S.J. (2014), “The appearance of creative behavior in later stage design processes”, *International Journal of Design Creativity and Innovation*, Vol. 2 No. 1, pp. 1–19.
- Stratonovich, R.L. (1960), “Conditional Markov Processes”, *Theory of Probability & Its Applications*, Vol. 5 No. 2, pp. 156–178.
- Tenison, C. and Anderson, J.R. (2016), “Modeling the distinct phases of skill acquisition.”, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 42 No. 5, pp. 749–767.
- Visser, W. (1994), “Organisation of design activities: Opportunistic, with hierarchical episodes”, *Interacting with Computers*, Vol. 6 No. 3, pp. 6–3.
- White, B., Blaylock, N. and Bölöni, L. (2009), “Analyzing Team Actions with Cascading HMM Team action analysis: Bounding Overwatch”, *FLAIRS Conference*.
- Wielinga, B. and Schreiber, G. (1997), “Configuration-design problem solving”, *IEEE Expert-Intelligent Systems and Their Applications*, Vol. 12 No. 2, pp. 49–56.