ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC 2018
August 26-29, 2018, Quebec City, Canada

DETC2018-85599

DESIGN STRATEGY TRANSFER IN COGNITIVELY-INSPIRED AGENTS

Ayush Raina
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213 USA
araina@andrew.cmu.edu

Christopher McComb
School of Engineering Design, Technology, and Professional Programs
The Pennsylvania State University
University Park, PA, 16802 USA
mccomb@psu.edu

Jonathan Cagan
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213 USA
cagan@cmu.edu

ABSTRACT

Planning and strategizing are essential parts of the design process and are based on the designer's skill. Further, planning is an abstract skill that can be transferred between similar problems. However, planning and strategy transfer within design have not been effectively modeled within computational agents. This paper presents an approach to represent this strategizing behavior using a probabilistic model. This model is employed to select the operations that computational agents should perform while solving configuration design tasks. This work also demonstrates that this probabilistic model can be used to transfer strategies from human data to computational agents in a way that is general and useful. This study shows a successful transfer of design strategy from human-to-computer agents, opening up the possibility of deriving high-performing behavior from designers and using it to guide computational design agents. Finally, a quintessential behavior of transfer learning is illustrated by agents while transferring design strategies across different problems, improving agent performance significantly. The work presented in this study leverages a computational framework built by embedding cognitive characteristics into agents, which has shown to mimic human problem-solving in configuration design problems.

1. INTRODUCTION

Design tasks like that of configuration design are highly complicated since the size of the feasible design space is exponentially large. Numerous problem-solving methods have been discussed that restrict the search process in these problems to a certain area of the design space using knowledge, heuristic, or other strategies [1–3]. Even though layout design is stated to be one of the simpler versions of configuration design problems, knowledge-based Artificial Intelligence techniques are needed to solve them efficiently as stated by Wielinga et al.[4]. This work centers on solving a layout problem using a team of cognitive agents that mimic human behavior [5]. The problem-solving method utilized by these agents is loosely based on the Propose Critique Modify methodology [6]. For an agent to follow this methodology, they must have a mechanism to produce new designs that can be proposed. There must be a verification method that can evaluate the current design to check if there have been any improvements in the quality. Then, a method to critique must identify the factors that led to the change in quality. This allows the final step of modification to target the precise reason and elevate or suppress that factor in order to produce higher quality designs in the future.

- Within the work introduced in this paper, proposing a new design is mostly based on design knowledge and heuristics that are learned and embedded in computational agents. This becomes a challenging task when computationally simulating designer behavior. By applying learned knowledge to a task, the design space can be reasonably constrained and solving the problem computationally becomes feasible. This work explores the role of design strategies in guiding this task of design generation based on previous knowledge. The term *design strategy* is used to refer to a policy, plan, or process heuristic [7] that a designer has for sequencing the operations used in solving a design problem. These strategies have been shown to have very distinct characteristics and can also be used to differentiate between high- and low-performance designer behavior. Design strategies can be emulated with probabilistic models; McComb, et al.[7,8], represent the design generation process as a stochastic operation selection task while Ji, et al. [9], model design preference probabilistically to represent design progression. Also, it has been shown that designers who perform well in configuration design tasks have strong spatial cognition abilities [10]. Since, these models used by McComb, et al., have been shown to capture the difference in operation sequencing for high- and low-performing designers, it can be said that these models represent spatial cognition abilities. This paper investigates the use of these as the primary form of representation of design strategies. This design strategy is also the primary factor for controlling the *propose* task of the agent since, operation selection defines what changes are made in the current design, while the *critique* and *modify* tasks are intrinsically controlled by the agent's learning framework (Subsection 2.2).

There are two questions that this paper aims to answer:

1. Can design strategies mined from designer behavioral data be utilized to augment computational agents?
2. Can a strategy learned for a specific problem be transferred over different problems?

This paper is divided into three content sections. The review of relevant literature is found in Section 2. Subsection 2.1 specifically explains the basic characteristics of the computational agents from the Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework that are employed to solve the design problems. A brief review of the importance of finding and learning sequences in engineering problems, specifically in engineering design, is given in Subsection 2.2. Then, details of the design study experiments which forms the basis for the computational experiments later on in the paper are given in Subsection 2.3. Section 3 introduces design strategies and their numerical representation as a probabilistic graphical model. Then different strategies are compared amongst each other to evaluate their relative performance. Finally, the section concludes with evaluating the transferability of these strategies across diverse problems. The paper ends with Section 4, which discusses the two major contributions of the paper and motivates the potential future work that could emanate from these results.

2. BACKGROUND

2.1 Agent-based Modeling of Design Teams Using CISAT

This paper explores the transfer of design strategies through the use of agent-based CISAT modeling framework. The CISAT model is composed of software agents that work together as a team to solve engineering problems. Each of these agents is analogous to an individual designer in a team who works in an iterative manner to solve engineering problems. CISAT implements eight cognitive team characteristics [5] which enables the agents to perform in a manner similar to an actual human design team. These characteristics represent a wide variety of cognitive behavior. For modeling the behavior of human designers in the experiments in later sections, the parameters have been kept to the default values as utilized by McComb et. al.[5]. However, this paper focuses on only one of the characteristics: operational learning. Operational learning deals with the process of learning how to select a design operation to modify the current solution. Prior efforts are explained in detail in the following subsection and also further advancements introduced in this work in the next section.

Although the focus of this work is the application of CISAT, there has been extensive use of agents to model design teams. These include process modeling [11,12] to simulate complex design tasks, mental modeling during team problem-solving with respect to both interaction structure [13] and agent memory [14], exploration of the effect of team structure and task complexity on the formation of transactive memory [15,16], improve managerial planning in product development [17], analyze adaptive team behavior in response to disruptions [18] and generative design via agent modeling [19,20].

2.2 Sequence-Learning Models for Design

Designers explore the design space in order to identify the best design in an iterative manner. At each iteration, the designer applies a certain operation that changes the design, and a combination of such operations may lead to a final design solution that satisfies the requirements for the design problem. Human designers can learn these sequences of operations over time and this learning is essential to human performance in a variety of domains [21]. Frequent use of sequential behavior while solving a task can also be indicative of expertise [22]. Markov Chain models have previously been utilized to represent the sequencing of design operations by designers for configuration design problems [8]. In a first-order Markov chain model, relations between the current operation and the immediately previous operation are captured. This relational dependency can be represented as a square matrix T which is k x k in size for each of the k different design operations that are defined in the problem. The matrix T and its probabilistic significance in design have been explained by McComb, et al. [8]. This Markov Chain model was later introduced in CISAT as a generative operation-selecting mechanism for the computational agents. During the design process, this model was also updated by real-time analysis of the operations. If, say, an operation j was taken after an operation i and it led to an increase in the quality of the solution, the element $T_{i,j}$ in matrix T shall be reinforced to a larger value giving it a higher precedence for being selected. These were compared with Multinomial Models to verify its effectiveness. Multinomial Model represents the selection of operations probabilistically in a non-sequential manner similar to a weighted die. There are k sides of this die where each side represents a design operation, so each operation has a probabilistic weight of occurrence. This model cannot represent a sequential relation in operations and was also seen to perform inferior to Markov Chains[23].

The Markov Chain study focused on sequencing short timeframe events like operations which are easily observable. Later work used Hidden Markov Models to represent not just these operations but also the underlying cognitive states that designers transition through. These states have a longer timescale and are also more abstract. The states potentially align with the different phases of design like conceptual design or detailed design. A Hidden Markov Model has multiple hidden states that can be time-dependent and each state can have distinct probability weights. They have been previously used for a plethora of applications ranging from speech recognition to protein modeling [24,25]. Some applications also pertain to problem-solving [26] and representing military tactics [27]. Hidden Markov Models can model a time-dependent preference of certain operations and hence work efficiently in representing design operation selection.

Hidden Markov Model were used to mine process heuristics from behavioral data for the design study of a house cooling system problem [7], indicating that designers transition across 4 states. Further, that work demonstrates that the top performing designers follow specific state-sequencing patterns that differ from those of low-performing designers. Hence, these models can represent high- and low-performing process knowledge about the design problem. That paper proposed that the process heuristics, which are the sequencing relations, can be generalized and extended to similar design problems and hence used as transferrable design strategies. We explore that possibility in the current work.

In this paper, two different approaches to learn operation sequencing will be used: offline and online learning. Offline learning means that designer data has been collected previously and is analyzed to extract observable designer behavior and sequencing patterns. This helps in creating a model that can replicate human designer results in design solving. This work builds on previous work by McComb, et al. [7], which uses Hidden Markov Models to represent human design behavior learned using the Baum-Welch algorithm [28]. Offline learning is analogous to how a designer can contemplate the decisions made during a previous design experience and then select the ones with favorable outcomes for future use. This process helps designers to differentiate high and low performing strategies and learn from mistakes which reinforce their knowledge for future problems. For the scope of this paper the process heuristics derived by a Hidden Markov Model have been used to simulate best and worst designer behavior and corresponding performance is evaluated in Subsection 3.2. Online learning, on the other hand, means real-time learning that occurs during the process of solving the design problem. The designer attempts different operations and then learns by noticing positive results. This, in theory, is analogous to reinforcement learning, where an agent tries different operations and identifies the best combination or sequence for achieving a given task and improves its policies based on the rewards it achieves. The agents in CISAT learn online using update rules based on a bipartite sequence recognition and generation process identified by Kotovsky and Simon for human participants solving the Thurstone Letter Series Completion task [29]. This updates and improves the process strategies in their operation selection models by reinforcing the probabilities that led to an increase in performance. This is an important feature of CISAT that makes the agents adaptive. Subsection 3.1 extends this online learning rule to Hidden Markov Models.

This paper extends the Hidden Markov Model that was used as a descriptive model [7] to be employed as a generative operation selection tool. The descriptive model was utilized to extract designer heuristics and characterize designer behavior. This work evaluates the use of a Hidden Markov Model in CISAT as an operation selection tool which can generate new designs. The Hidden Markov Model provides a rich hierarchical representation of probabilistic dependence of states and operations which governs the behavior of the agent. More about utilizing Hidden Markov Model as a generative operation selection model is given in Subsection 3.1.

2.3 Design of Internet-Connected Home Cooling Systems

This work utilizes data collected in a design study [30] conducted to analyze behaviors of human designers during a team-based design process. Participants were given the task of designing an internet-connected system for maintaining the temperature within a home. While solving, each participant was given access to a graphical user interface that facilitated design and evaluation as well as recording every operation taken by the participants. The participants solved the design problem in three-person teams. Each team was introduced to the problem by showing them the floor layout and giving them a set of design operations to utilize. The teams were given a Medium sized room layout (with 13 rooms, shown in Figure 1). The target was to minimize the peak temperature inside the house and limit it to 24°C. The cost of the system was an additional factor which needed to be minimized and was preferably limited to $20,000. The students had 50 iterations of operations available to achieve the target temperature while minimizing the overall cost. The graphical interface allowed them to take one operation at a time and get real-time feedback for their selection of operation. Participants could use three component types to construct their solutions: a cooler to channel air into the rooms to produce the cooling effect, a sensor to sense the temperature of the room, and a processor that would control the coolers based on the temperature data collected from the sensors. Nine design operations were available based on adding, deleting, moving and optimizing the components. At the end of each study, the data collected is a sequence of design operations taken by the design team in order to achieve the objective.

For evaluating the solution, the mean temperature in each room was computed using principles of heat and mass transfer as coolers and sensors were added during the simulation. Peak temperature and total cost were computed based on the simulation results. Peak temperature is the highest temperature obtained in any room in the house during the simulated time period. Total cost is the sum of the cost of the products in the system and their projected 10-year operating cost. For evaluation in this study, we define normalized cooling cost as the ratio of the drop in peak temperature (from initial value) to total cost as a performance metric.

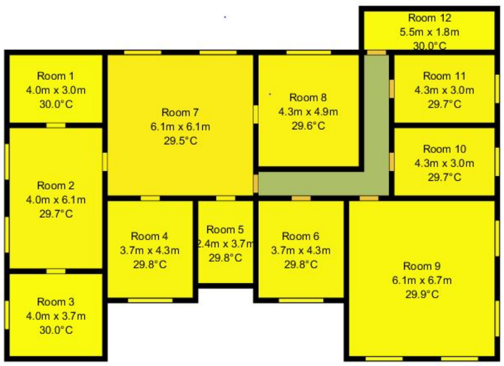

FIGURE 1. MEDIUM SIZE HOUSE LAYOUT.

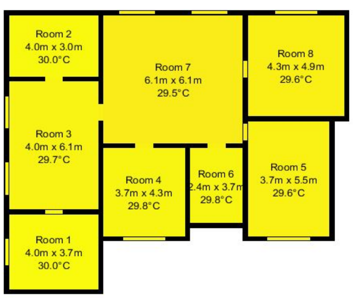

FIGURE 2. SMALL SIZE HOUSE LAYOUT.

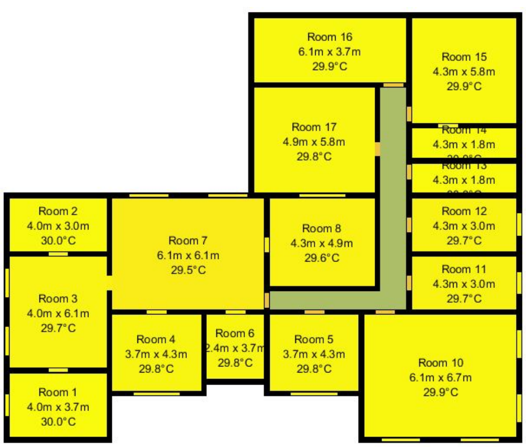

FIGURE 3. LARGE SIZE HOUSE LAYOUT.

Although the study was conducted in teams, in this work the problem is considered as a single designer task. This data was previously analyzed with the same assumption to derive design strategies by estimating probabilistic graphical models [7,8]. Figures 2 and 3 are the other variants of the problem where the house size is changed. These problems are used to evaluate the effectiveness of transfer learning in Subsection 3.3.

3. INVESTIGATING STRATEGY TRANSFER IN CISAT

This section investigates how design strategies can be transferred across problems and explain the process required to do so. The term design strategy can also be called process heuristics [7]. In order to transfer learned strategies, the first step is efficient modeling and representation of those strategies. Subsection 3.1 answers that need by showcasing a method which was previously used as a descriptive model to mine process heuristics [7], but is extended to be utilized as an adaptive operation selection model with features that are complex enough to capture time-dependent operation sequencing. Multiple operation selection models are being compared that can represent designer strategies of CISAT agents. Also, computational experiments are conducted, and their results are compared to check the effectiveness of the model. Later, the parameters that control the strategies that the agent follows are identified. It also explains the use of online learning of strategies that improve them in real time. Subsection 3.2 enlists the various strategies that have been extracted from human designer data and explains the significance of each of them. Later, a preliminary comparison is made that justifies the intent of using an experiment behind creating some of the strategies. Finally, in Subsection 3.3, the relative performance of strategies is compared over different problems and final results for strategy transfer are discussed. This idea of transferring knowledge from one problem to another to improve performance is analogous to the concept of transfer learning [31]. It also explains the importance of operation sequencing in design problems and motivates the need for a robust operation selection model that can represent adaptable design strategies.

3.1 Hidden Markov Model as a framework for operation selection

CISAT has previously used probabilistic Markovian models to guide agent operations at every iteration in the design process. As shown by McComb, et al. [8], Markov Chain models with online learning were efficient in learning design sequences and helped the agents perform similarly to the human designer in the design activity. Hidden Markov Models provide more structured information about the design process than just Markov chains. Here the next operation is dependent on the hidden state of the current operation [7]. These state-based preferences portray a behavior similar to how designers have different strategies during sequential phases of design, this shows an example of opportunistic design where designers adapt their process according to the emerging strategies [32]. However, Hidden Markov models have not been integrated into CISAT to guide operation selection. This subsection explains how Hidden Markov Models are used to make the decision at every time step in CISAT and also how online learning is implemented to make it a dynamic learning model.

A Hidden Markov Model has emission vectors representing probabilities of operation selection for each of the states and also state transition probabilities. Our model has 9 operations and 4 hidden states. The number of hidden states was determined in previous work by optimizing the log likelihood of the model [7]. The transition matrix T is 4x4 matrix since the number of states is 4 and there is a chance for the agent to move to any other state from a given state. The emission matrix, E, stores the weights for selecting an operation given a particular state; hence, it is a 4x9 matrix. Throughout the design process, it is these two matrices that govern operation selection, playing a major role in defining the performance of the agent and the team as a whole. This model has 44 degrees of freedom (12 for the transition matrix and 32 for the emission matrix) and represents the process in a two-step hierarchical manner. The weights of the matrices along with the online learning process which updates these weights determine the strategy that the agents employ to solve the design process and for the same reason have been referred to as design heuristics [7]. This work also shows an example of such strategy for configuration design problems, where the good-performing designers have a specific weight distribution in these matrices that correlates to designers using topological design operations early in the process and then spending most of the later time in optimizing the components.

T_{ij} refers to the probability of transitioning from state i to state j. Similarly, the emission matrix represents the probabilities of choosing a particular operation given a state; E_{ik} refers to the probability of choosing a particular operation k given a state i. The operation selection process in a Hidden Markov Model is a 2-step process. At every iteration, the agent stochastically chooses the next state from the current state row in T_{ij} and then stochastically chooses the operation based on the new state from E_{ik}. So, the higher value of the respective probability makes that state or operation more preferable. This operation selection model is also dynamically changing its weights as it learns at every iteration. The online learning method that is employed for the Hidden Markov Model is similar to which was used previously for Markov Chains for operation sequencing by McComb et al. [5]. It works on collecting real-time reward based on the change in quality and then proportionally reinforces the probability weights of the states and operation that led to that reward. So, if state s leads to a state s' and an agent selects an operation a which causes a change in the quality of the solution, then the reward updates the weights by the operations shown in Equations 1 – 4:

$$T_{ss'} = T_{ss'} \cdot (1 + \Delta \cdot lr) \quad (1)$$

$$T_{ss'} \rightarrow Normalize(T_{ss'}) \quad (2)$$

$$E_{s'a} = E_{s'a} \cdot (1 + \Delta \cdot lr) \quad (3)$$

$$E_{s'a} \rightarrow Normalize(E_{s'a}) \quad (4)$$

The variable $T_{ss'}$ refers to the transition probability from s to s' and $E_{s'a}$ refers to the emission probability of operation a given a state s'. The variable Δ refers to the change in the quality of design; it could be both negative as well as positive. The variable lr is the learning rate which determines how greedily the probability is changed in the process. This update in weights increases the probability of beneficial transitions and emissions in future iterations. This online learning method is based on the method previously utilized in CISAT [23] and has been adapted here to be used with Hidden Markov Models to maintain uniformity when comparing different operation selection models in CISAT. Standard algorithms like Baum-Welch [33], are not stable for online learning in their original formulation [34] and significant advancements [35,36] need to be implemented for application in an online setting, an area to be explored in future work.

CISAT simulations were used to compare McComb, et al.'s Multinomial and Markov Chain [5,23] operation selection models to the Hidden Markov Model proposed in the current work. The results are based on normalized cooling cost achieved in the design process and are shown in Figure 4. The weights in the operation selection models are initialized randomly for each of these three models. It can be observed that the Multinomial Model which has only 8 degrees of freedom performs the worst amongst all three as expected from the previous results [8]. Both Markov chains and Hidden Markov Models perform equally well throughout the design process, even though Markov chains have more degrees of freedom than Hidden Markov Models (72 versus 44). Additionally, a Hidden Markov Model allows an extra time-based state variable in model properties that link operation emission probabilities to the state that the model is in. This enables the model to capture hierarchical information and efficiently represent a time-based design strategy which cannot be achieved by just using Markov Chains. Hence, compared to the Multinomial model and Markov Chain model previously used in CISAT, Hidden Markov Models are a preferred choice in being used as an operation selection model for CISAT since they are simpler with fewer variables while being able to represent rich hierarchical relationships required for efficient operation selection.

3.2 Using Problem-solving strategies to guide operation selection

As previously discussed, the weight distribution of the state transition and emission matrices along with the online learning mechanism together represent the strategy that a designer undertakes during the design process. This subsection explores effects of having weight distributions that are learned from previous human studies on the performance of computational agents. To simulate the experiment, the matrices are introduced as a part of the operation selection model in CISAT. The matrices are the same for all agents that are part of a single team in CISAT, meaning that all agents of a team follow the same design strategy

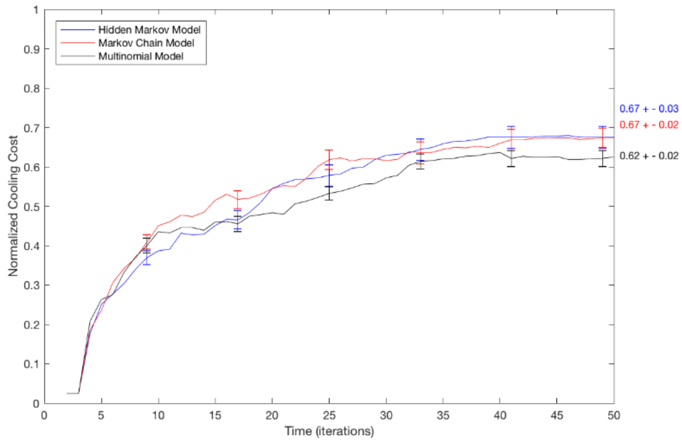

FIGURE 4. COMPARISON OF OPERATION SELECTION MODELS IN CISAT. THE ERROR BAR REPRESENTS ±1 STANDARD ERROR.

for solving the problem. Five sets of different weight distributions each representing a different design strategy were used to initialize the values for these matrices. Hence, Figure 5 shows the values with which the model was initialized for the different strategies. The color coding represents high probability values as white (1.0) or yellow and low probability values as black (0.0) or brown, while intermediate values are shown with different shades of red. It must be noted that the values represented in Figure 5 are only the initial values for these strategies since they get updated at each iteration due to online learning. The difference in these design strategies and their resultant effect only occur because of different initialization of the weight distributions; all these sets follow the same online learning mechanism as explained before. The description of these strategies is as follows:

1. *Best and Worst Problem Specific:* The weight distributions in this set were extracted from a human design study using an offline learning method by McComb et al. [7]. The study was conducted on the same cooling problem as mentioned in Subsection 2.3. The Best Problem Specific strategy represents the strategies that high performing designers used in the design study while the Worst Problem Specific represents low performing designers' strategy. These have been named Problem Specific since these are exact values from the design study and could be valid only for the original problem from which they are derived.

2. *Best and Worst Heuristic:* Heuristic strategies are an approximation to the Problem Specific strategies and capture the relative values of probabilities across a state. The way these matrices have been created is by initializing the matrix to insignificant but non-zero values. Then large values were added to the elements corresponding to the highly weighted elements in the Problem Specific strategy. Finally, the values were normalized to convert into probabilities. So, all elements have a non-zero weight which makes it different from the Problem Specific set. This increases the randomness in the model since an agent applying this strategy can explore the non-preferred operations. Hence, this approximation allows the capture of features from the Problem Specific set while preventing the matrices from being too specific (overfitting the data).

3. *Random Initialization:* Here both the state transition and emission matrices are initialized with random probabilities. This can give a preference to any state pair for transition and state-operation pair for emission hence formulating a random strategy for operation selection in the initial phase. This strategy represents a computational agent that begins the design process by selecting random operations but eventually attributes high probabilities to high performing operations over time due to online learning.

An experiment was conducted to compare the performance of 5 different strategies: Best Heuristic, Worst Heuristic, Best Problem Specific, Worst Problem Specific and Random Initialization. CISAT was used to simulate the problem-solving process of a design team. Apart from the different strategies, all the team characteristics were kept constant for the experiment. This run was conducted on a medium size house layout which was also the original design problem used in the study to extract designer strategies for formulating the Problem Specific and Heuristic strategies. Results are shown in Figure 6, representing how the efficiencies for different sets change over time, also the end efficiencies for each of the strategies are represented by their corresponding variance numerically in a decreasing order.

The results in Figure 6 show a distinct difference between human data-derived strategies (Problem Specific and Heuristic) and the Random Initialization strategy. This highlights the importance of the operational sequencing skills that human designers use to solve a problem since Problem Specific and Heuristic strategies have operation sequencing preferences which lead to better results throughout the design process. These high-performance results cannot be replicated by using the random initialization strategy, hence, reinforcing the understanding that human strategies do exhibit high performing designer behavior.

These results provide evidence for two findings. First, the heuristic strategies show similar trends to that of problem specific strategies, verifying successful extraction of important features. This can be said because the best heuristic performs very close to the best problem specific strategy while the worst ones are also close to each other. Second, since there is a significant gap between the random initialization strategy and the human-derived inputs (both heuristic and problem specific); this illustrates the importance of using sequencing knowledge to perform better in design tasks. This is an example where strategies were successfully transferred from humans to machines (agents) through the use of these probabilistic computational models. It can be observed in Figures 5c and 5d that the non-highlighted regions are brown in comparison to the non-highlighted regions in Problem Specific case, Figures 5a and 5b which are black. The small, non-zero probabilities of selecting these operations in heuristic strategies allows the agent to explore different states and operation and can eventually better improve their existing strategies by online learning. This motivation to explore may help the agent to experiment and determine if a previously unfavorable operation is useful in a new scenario, allowing agents to behave more stochastically and possibly improve learning.

3.3 Design strategy transfer

Previous subsections have demonstrated that the performance of a CISAT team improves considerably when the strategy of high performing designers is used for solving the same problem, even though further learning takes place during problem solving. This subsection tests the performance of the CISAT teams when the problem being solved is similar but not identical to the problem on which the strategies were learned. This investigation has goals similar to the concept of transfer learning in reinforcement learning [37]. Transfer learning is an approach used to obtain pre-trained values of policies (strategies) that are trained on a specific problem and then used for a new but similar problem. Successful strategy transfer should increase performance compared when using just a random initialization strategy. The experiment uses *small* and *large* variants of house layouts in this home cooling system design problem. The difference between them is in terms of the complexity of the problem; as the number of rooms and area increases in a house, designing a cooling system becomes increasingly difficult. But they are all part of the same family of problems since these problems are similar to the medium size house layout. The designers still need to create a cooling system for the house and have same operations and states for solving each one of them. The performance of CISAT is assessed on these problems using strategies learned for the medium house layout mentioned in Subsection 3.2. The results are discussed in the following paragraphs.

Torrey and Shavlik [31] define three main characteristics that show a successful transfer of strategy. First, the initial performance of an agent using a previous strategy is significantly higher than a randomly initialized agent. This is represented by the initial slope of the performance versus iteration graph. Second, the time an agent takes to fully learn the target task reduces compared to the time needed without a transferred strategy. Here, the performance may flat line earlier than other agents showing that it has learned the specific policy to solve the problem. Third, the final efficiency achieved is higher for agents with learned strategies than for agents who are learning everything from scratch. This subsection presents computational experiments that compare the different strategies transferred from the medium house and uses certain measures [31] to compare their performance for successful transfer of design strategy.

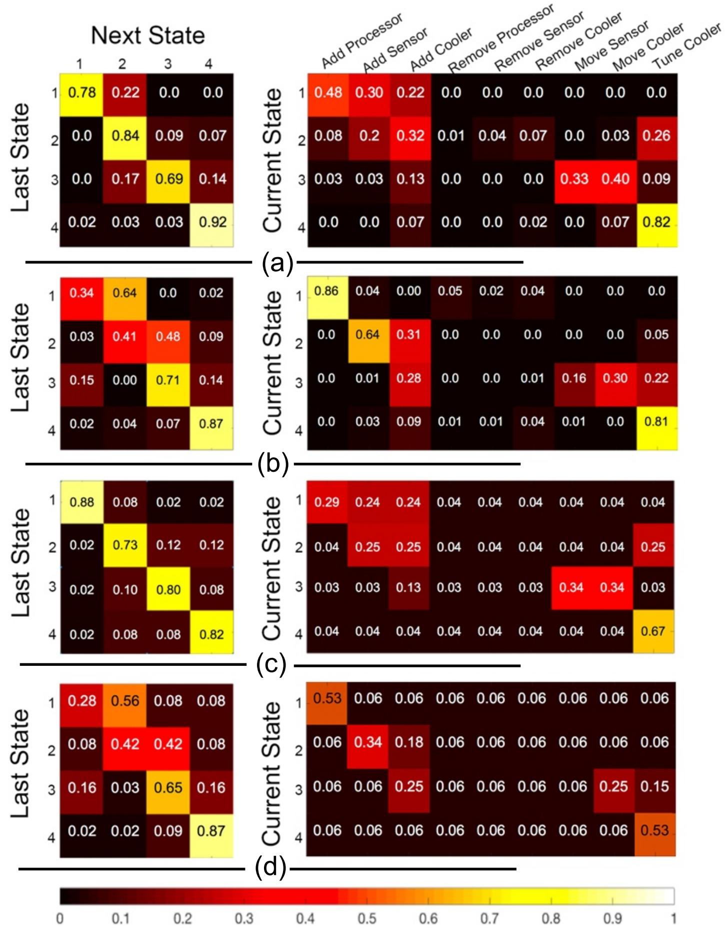

FIGURE 5. STATE TRANSITION MATRICES (LEFT) AND EMISSION MATRICES (RIGHT) FOR FOLLOWING STRATEGIES: (A) BEST PROBLEM SPECIFIC, (B) WORST PROBLEM SPECIFIC, (C) BEST HEURISTIC, AND (D) WORST HEURISTIC. COLOR CODING INDICATES PROBABILITY OF SELECTION.

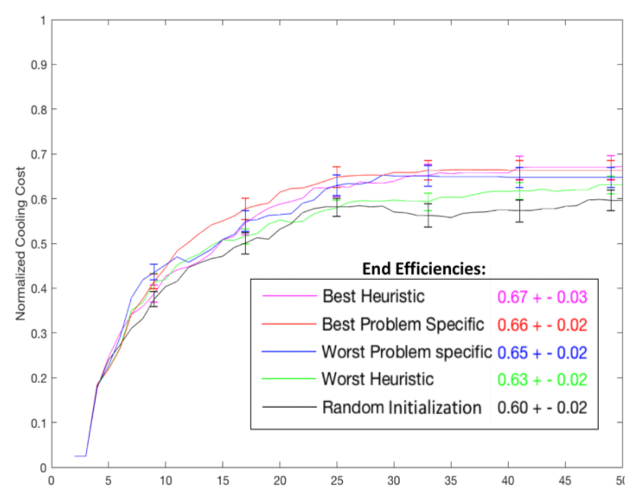

FIGURE 6. DESIGN STRATEGY COMPARISON FOR MEDIUM SIZE HOUSE LAYOUTS. THE ERROR BAR REPRESENTS ±1 STANDARD ERROR.

For the small size house, the results are shown in Figure 7. The design solving process is computationally less intensive since this layout has fewer rooms. This is also verified in the plots, as different strategies show very similar final efficiencies; the difference in strategies is not a significant factor to determine the final efficiency. However, a higher gradient for efficiency increase can be observed for strategies derived from human data (both problem specific and heuristic) as compared to the random initialization strategy enabling them to achieve a high-efficiency value very early in the process. The best heuristics and best problem specific strategies perform better than the rest, which has been a consistent result for all the experiments. The random initialization strategy shows poor efficiency values in the beginning but a high final efficiency; this is due to online learning improving the initial bad strategy over time for this lower complexity problem. The large house is a more complicated design problem than both the small and the medium house since there are more number of rooms and a larger area; performance results for it are shown in Figure 8. Here the best heuristics and best problem specific strategies perform considerably better than worst heuristics and worst problem specific, respectively. Overall the best performance was achieved by the Best Heuristics; the difference between best heuristics and best problem specific is significant in this case showing that a generalized human design strategy can perform equal or better than problem specific strategies over a range of problems. In this case, the Random Initialization strategy performs very low as compared to the human inputs (both problem specific and heuristic).

The results in Figures 7 and 8 provide promising evidence for successful transfer of strategies over different problems in the same problem domain. The efficiency plots show similar trends throughout for these different strategies: The best heuristic and problem specific strategies always have a higher initial efficiency compared to the random initialization strategy; they fully learn the strategy faster (shown by a flat line), and also have a higher end efficiency. Hence, we can conclude that our results provide a successful example of transfer learning in accordance with Torrey and Shavlik [31].

The Best Heuristics worked the best for all the three different sizes of the house. This shows that the method used to generate these heuristic strategies from problem specific strategies is an efficient way to capture the best designer behavior and generalize it for its usage across problems. Heuristic strategies have an increased randomness in the values as compared to problem specific strategies which lead to better generalization and applicability over a range of problems as shown by slightly higher end efficiency results. Having non-zero probabilities for seemingly non-optimal operations allows the model to explore applying these operations and reinforce them if favorable results are obtained. This property of Best Heuristics makes it the best available strategy for transfer learning since they are robust and have consistently been performing more than or at par with the best problem specific inputs across a variety of problems.

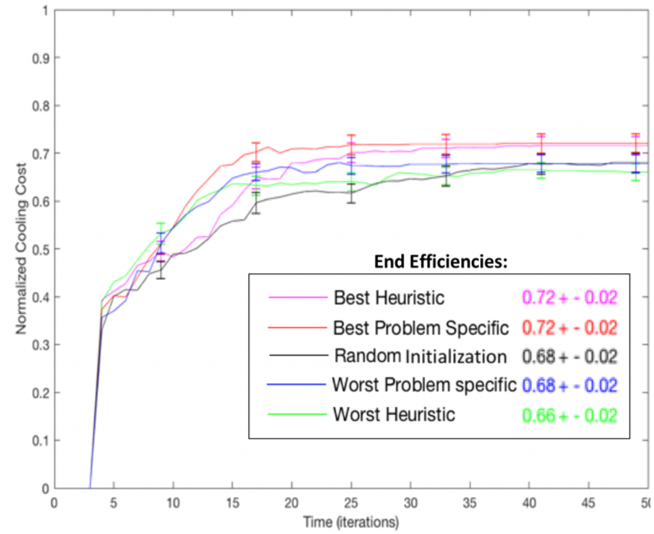

FIGURE 7. DESIGN STRATEGY COMPARISON FOR SMALL SIZE HOUSE LAYOUTS. THE ERROR BAR REPRESENTS ±1 STANDARD ERROR.

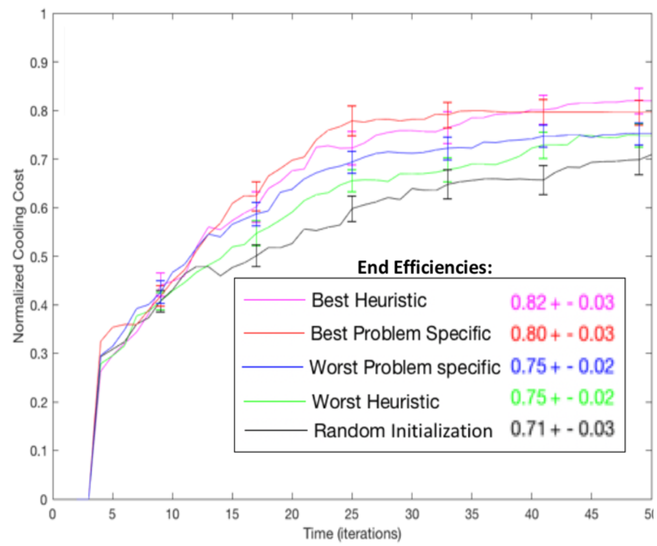

FIGURE 8. DESIGN STRATEGY COMPARISON FOR LARGE SIZE HOUSE LAYOUT. THE ERROR BAR REPRESENTS ±1 STANDARD ERROR.

4. CONCLUSION AND FURTHER DISCUSSION

This work explores open questions relevant to the agent-based computational design and strategy transfer through the CISAT modeling framework. The paper explores the representation of design strategies as a Hidden Markov Model, their application to engineering design problems, evaluation of their generality, and, finally, how they could be transferred across similar problems. Also, the observations made in Section 3 mention some ideas about how these problems are replicable for both human and machine implementations for future usage. Broadly speaking, all the findings of this paper can be summarized in two major results that address some of the open questions in the field of computational design and can guide future work in this domain. First, the work demonstrates the successful transfer of design strategies from human designers to computational agents. The observations made in Subsection 3.2 about design strategies from human designer data show similar performance trends even when applied by computational agents. Also, the performance values of best and worst strategies (both heuristic and problem-specific) follow the same relative performance trend. This gives significant evidence that human design heuristics can be represented through these probabilistic models. These models also establish a common ground between human designers and computational agents when representing design strategies, and thus opens up new avenues of research relevant to design methodology development for hybrid human-agent design teams.

Second, this work demonstrates the ability to achieve transfer learning in agent-based models through state-based probabilistic models. These models leverage past human experiences using offline learning to augment the abilities of computational agents, and thus open up a direction of research that merges concepts of transfer learning with engineering design. The results and experiments conducted in Subsection 3.3 begin to explore this intersection. The strategies are tested on three different problems and the Best Heuristic strategy which is learned from a medium size house layout shows the best performance in all three problems. The increased agent performance on different problems signifies that using previous experience in the form of design strategies helps in performing better. These inferences illustrate how previous design experiences could be used to develop generalizable strategies and improve performance in relevant design tasks in the future. This is clearly shown in this work across different problems within a similar domain; how far these strategies transfer into other domains will be explored in future work.

Ultimately, this work demonstrates a means of representing strategies that can be extracted from humans and applied by computational agents. These strategies preserve the behavior of high-performing designers as well as considerably boosting performance when transferred across problems.

ACKNOWLEDGEMENTS

This material is based upon work supported by the Defense Advanced Research Projects Agency through cooperative agreement No. N66001-17-1-4064. 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

[1] Puppe, F., 1993, *Systematic Introduction to Expert Systems: Knowledge Representations and Problem Solving Methods*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, NJ, USA.

[2] Maher, M. Lou, 1990, “Process Models for Design Synthesis,” AI Mag., **11**(4), p. 49.

[3] Breuker, V. de velde, 1994, *Common KADS Library for Expertise Modelling*, IOS Press, Amsterdam, The Netherlands, The Netherlands, The Netherlands, The Netherlands.

[4] Wielinga, B., and Schreiber, G., 1997, “Configuration-Design Problem Solving,” IEEE Expert. Syst. their Appl., **12**(2), pp. 49–56.

[5] McComb, C., Cagan, J., and Kotovsky, K., 2015, “Lifting the Veil: Drawing Insights about Design Teams from a Cognitively-Inspired Computational Model,” Des. Stud., **40**, pp. 119–142.

[6] Chandrasekaran, B., 1990, “Design Problem Solving: A Task Analysis,” AI Mag., **11**(4), pp. 59–71.

[7] McComb, C., Cagan, J., and Kotovsky, K., 2017, “Mining Process Heuristics From Designer Action Data via Hidden Markov Models,” J. Mech. Des., **139**(11), p. 111412.

[8] McComb, C., Cagan, J., and Kotovsky, K., 2017, “Capturing Human Sequence-Learning Abilities in Configuration Design Tasks Through Markov Chains,” J. Mech. Des., **139**(9), pp. 91101–91112.

[9] Ji, H., Yang, M. C., and Honda, T., 2007, “A Probabilistic Approach for Extracting Design Preferences From Design Team Discussion,” (48043), pp. 297–306.

[10] Kim, M. J., and Maher, M. Lou, 2008, “The Impact of Tangible User Interfaces on Spatial Cognition during Collaborative Design,” Des. Stud., **29**(3), pp. 222–253.

[11] Jin, Y., Levit, R., and Levitt, R., 1996, “The Virtual Design Team : A Computational Model of Project Organizations,” Comput. Math. Organ. Theory, **2**(3), pp. 171–196.

[12] Olson, J., Cagan, J., and Kotovsky, K., 2009, “Unlocking Organizational Potential: A Computational Platform for Investigating Structural Interdependence in Design,” J. Mech. Des., **131**, pp. 031001-1–13.

[13] Dionne, S. D., Sayama, H., Hao, C., James, B., and Bush, B. J., 2010, *The Role of Leadership in Shared Mental Model Convergence and Team Performance Improvement: An Agent-Based Computational Model*, Elsevier Inc.

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

[15] Singh, V., Dong, A., and Gero, J. S., 2012, “Computational Studies to Understand the Role of Social Learning in Team Familiarity and Its Effects on Team Performance,” CoDesign, **8**(1), pp. 25–41.

[16] Singh, V., Dong, A., and Gero, J. S., 2013, “Social Learning in Design Teams: The Importance of Direct and Indirect Communications,” Artif. Intell. Eng. Des. Anal. Manuf., **27**(2), pp. 167–182.

[17] Perišić, M. M.; Martinec, T.; Štorga, M.; Kanduč, T., 2016, "Agent-Based Simulation Framework to Support Management of Teams Performing Development Activities," *14th International Design Conference (DESIGN 2016), 16-19.06.2016.,* Dubrovnik, Hrvatska, pp. 1925–1936.

[18] Perišic, M. M., Štorga, M., and Gero, J., 2017, "Building a Computational Laboratory for the Study of Team Behaviour in Product Development," *Proceedings of the International Conference on Engineering Design, ICED*.

[19] Campbell, M. I., Cagan, J., and Kotovsky, K., 1999, "Engineering Design A-Design : An Agent-Based Approach to Conceptual Design in a Dynamic Environment," Res. Eng. Des., (1999), pp. 172–192.

[20] Orsborn, S., and Cagan, J., 2009, "Multiagent Shape Grammar Implementation: Automatically Generating Form Concepts According to a Preference Function," J. Mech. Des., **131**(12), p. 121007.

[21] Clegg, B. A., DiGirolamo, G. J., and Keele, S. W., 1998, "Sequence Learning," Trends Cogn. Sci., **2**(8), pp. 275–281.

[22] Pretz, J. E., Naples, A. J., and Sternberg, R. J., 2003, "Recognizing, Defining, and Representing Problems," *The Psychology of Problem Solving*, pp. 3–30.

[23] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks," *Design Computing and Cognition '16*, pp. 401–418.

[24] Jelinek, F., Bahl, L., and Mercer, R., 2006, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," IEEE Trans. Inf. Theor., **21**(3), pp. 250–256.

[25] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D., 1994, "Hidden Markov Models in Computational Biology: Applications to Protein Modeling," J. Mol. Biol., **235**(5), pp. 1501–1531.

[26] R., A. J., and M., F. J., 2013, "Discovering the Sequential Structure of Thought," Cogn. Sci., **38**(2), pp. 322–352.

[27] White, B. A., Blaylock, N., and B□l□ni, L., 2009, "Analyzing Team Actions with Cascading HMM.," *FLAIRS Conference*.

[28] Baum, L. E., Petrie, T., Soules, G., and Weiss, N., 1970, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," Ann. Math. Stat., **41**(1), pp. 164–171.

[29] Simon, H. A., and Kotovsky, K., 1963, "Human Acquisition of Concepts for Sequential Patterns," Psychol. Rev., **70**(6), pp. 534–546.

[30] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Data on the Configuration Design of Internet-Connected Home Cooling Systems by Engineering Students," Data Br., **14**, pp. 773–776.

[31] Torrey, L., and Shavlik, J., 2009, "Transfer Learning," Mach. Learn., pp. 1–22.

[32] Guindon, R., 1990, "Designing the Design Process: Exploiting Opportunistic Thoughts," Human-Computer Interact., **5**(2–3), pp. 305–344.

[33] Baum, L. E., Petrie, T., Soules, G., and Weiss, N., 1970, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," Ann. Math. Stat., **41**(1), pp. 164–171.

[34] Baldi, P., and Chauvin, Y., 1994, "Smooth On-Line Learning Algorithms for Hidden Markov Models," Neural Comput., **6**(2), pp. 307–318.

[35] Chis, T., and Harrison, P. G., 2015, "Adapting Hidden Markov Models for Online Learning," Electron. Notes Theor. Comput. Sci., **318**, pp. 109–127.

[36] Baldi, P., and Chauvin, Y., 1994, "Smooth On-Line Learning Algorithms for Hidden Markov Models," Neural Comput., **6**(2), pp. 307–318.

[37] Taylor, M. E., and Stone, P., 2009, "Transfer Learning for Reinforcement Learning Domains : A Survey," J. Mach. Learn. Res., **10**, pp. 1633–1685.

