Transferring Design Strategies From Human to Computer and Across Design Problems

Ayush Raina
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: araina@andrew.cmu.edu

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

Christopher McComb
School of Engineering Design, Technology,
and Professional Programs,
The Pennsylvania State University, University Park,
State College, PA 16802
e-mail: mccomb@psu.edu

Solving any design problem involves planning and strategizing, where intermediate processes are identified and then sequenced. This is an abstract skill that designers learn over time and then use across similar problems. However, this transfer of strategies in design has not been effectively modeled or leveraged within computational agents. This note presents an approach to represent design strategies using a probabilistic model. The model provides a mechanism to generate new designs based on certain design strategies while solving configuration design task in a sequential manner. This work also demonstrates that this probabilistic representation can be used to transfer strategies from human designers to computational design agents in a way that is general and useful. This transfer-driven approach opens up the possibility of identifying high-performing behavior in human designers and using it to guide computational design agents. Finally, a quintessential behavior of transfer learning is illustrated by agents as transferring design strategies across different problems led to an improvement in agent performance. The work presented in this study leverages the Cognitively Inspired Simulated Annealing Teams (CISAT) framework, an agent-based model that has been shown to mimic human problem-solving in configuration design problems. [DOI: 10.1115/1.4044258]

Keywords: agent-based design, design automation, generative design

1 Introduction

Humans are capable of utilizing prior experiences, previously acquired knowledge, or learned concepts from previous experiences to enhance their performance at new but similar tasks. This concept is called transfer of learning [1] and is an abstract skill that humans possess, making them fast learners and efficient problem solvers. They are seen to attain expert skills with growing experience; hence, expertise and experience are often interchangeably used to refer to a high skill level in humans [2]. Even though this process of transfer of learning is of widespread use in humans, modeling this phenomenon using computational agents has been a great challenge for researchers. This could be a useful skill for computers, especially since machines do not tire and can accomplish certain tasks cheaper and more efficiently than humans [3]. This motivates the need to develop agents that can learn from past experience and solve new problems. This work presents an approach that represents design strategies as a probabilistic model and successfully transfers learned strategies from human designers to computational design agents.

This work specifically focuses on configuration design as a sequential decision-making process, and configuration design is a category of design tasks where numerous components need to be selected and assembled in order to accomplish an objective [4]. These are highly complicated design problems since the size of the feasible design space is dependent on all the possible locations/configurations of each component and searching in this space could be computationally expensive to solve. Numerous problem-solving methods have been discussed that restricts the search process in these problems to a certain area of the design space using knowledge, heuristics, or artificial intelligence [5–8]. This work explores the role of design strategies in guiding this process of solving a configuration design problem using a team of cognitive agents that mimic human behavior [9]. The term *design strategy* is used to refer to a policy, plan, or process heuristic [10] that a designer uses for sequencing the operations used in solving a design problem. The problem-solving method utilized by these agents is loosely based on the propose critique modify methodology [11]. The *propose* task in this case is controlled by the strategy model, governing how an agent selects sequential design operations. The *critique* and *modify* tasks are intrinsically controlled by the agent's learning framework (explained in Sec. 2.2) that evaluates design operation and updates its strategy.

Design processes have been previously emulated with probabilistic models [10,12,13]; models used by McComb et al. [10] were shown to capture the difference in high- and low-performing designers, capturing parametric details relevant to the design knowledge embedded in human designers. This paper builds upon that work by adapting hidden Markov models (HMMs) as a generative model representing the embedded strategies for computational agents as its parameters. Different variations of these strategies are generated from human data, which are used as priors for the model parameters. Finally, these strategy priors are used to solve new unseen problems evaluating their generality and transferability of the strategy models across problems. There are two questions that this work aims to answer:

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

This note is divided into three content sections. The review of relevant literature and the design study is found in Sec. 2. Section 3 introduces design strategy representation as a probabilistic graphical model, and then, the performances of different strategies are compared, and, finally, it concludes with evaluating the transferability of these strategies across problems. The note ends with Sec. 4, which discusses the two major contributions of the paper and motivates potential future work.

2 Background

2.1 Agent-Based Modeling of Design Teams Using CISAT.

This paper explores the transfer of design strategies using the Cognitively Inspired Simulated Annealing Teams (CISAT) framework [9]. 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 working in an iterative manner. CISAT implements eight cognitive team characteristics, which enable the agents to perform in a manner

Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received March 16, 2019; final manuscript received July 5, 2019; published online July 17, 2019. Assoc. Editor: Ying Liu.

similar to an actual human design team. Model parameters have been kept to the default values for experiments in later sections [9]. One of the embedded characteristics is operational learning, which deals with the process of learning how to sequence design. Further advancements to this framework are discussed in the subsequent sections. Although the focus of this work involves only the application of CISAT, there has been extensive prior use of agents in modeling design teams. These include process modeling [14,15] and mental modeling [16,17] for solving design tasks, exploration of the effect of team structure and task complexity on the formation of transactive memory [18,19], improved managerial planning in product development [20], analyzed adaptive team behavior [21], and finally generative design [22,23].

2.2 Sequence-Learning Models for Design.

Design can be considered as a sequential decision-making process [24] where designers explore design spaces in an iterative manner. Human designers can learn these sequences of operations over time, which is essential to human performance in a variety of domains [25]. This sequential learning ability and its application to generate designs have been simulated by using probabilistic models such as a Markov chain (MC) model [12]. The first-order MC model can capture the relation between consecutive design operations and represent short timeframe events. These MC models have also been used to generate new designs [26] and were seen to function similar to humans. Later work used HMMs to represent higher level operation sequencing in design processes. HMMs have abstract hidden states that were shown to potentially align with the different phases of design like conceptual design or detailed design [10]. This paper extends the HMM that was used as a descriptive model [10] to be employed as a generative operation selection tool in CISAT. HMMs can model a state-dependent preference of design operations and hence can be used to perform operation selection in computational agents. The HMM provides a rich hierarchical representation of probabilistic dependence of design operations on the state of the agent, which allow the agents to change their behavior as states change with progressing iterations (equivalent to time). More about utilizing HMM as a generative operation selection model is given in Sec. 3.1.

2.3 Design of Internet-Connected Home Cooling Systems.

This work utilizes the data collected in a human team design study by McComb et al. [27]. Participants were given the task of designing an Internet-connected house cooling system. Each participant was given access to a graphical user interface and evaluation metrics. The participants solved the design problem in three-member teams. The teams were provided a medium-sized room layout (with 13 rooms, as shown in Fig. 1). The target was to limit the peak temperature within 50 iterations (or changes to the design). The cost of the system was an additional factor, which needed to be minimized. Nine design operations were available based on adding, deleting, moving, and optimizing the cooling system components. For evaluating the solution, the mean temperature in each room was computed using principles of heat and mass transfer. Peak mean 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, the normalized cooling cost is defined as the ratio of the drop in peak temperature (from initial value) to total cost as a performance metric. Although the study was conducted in teams, individual designers' moves were captured, and in this work, the problem is considered as a single designer task. These data were previously analyzed with the same assumption to derive design strategies by estimating probabilistic graphical models [10,12]. 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 Sec. 3.3.

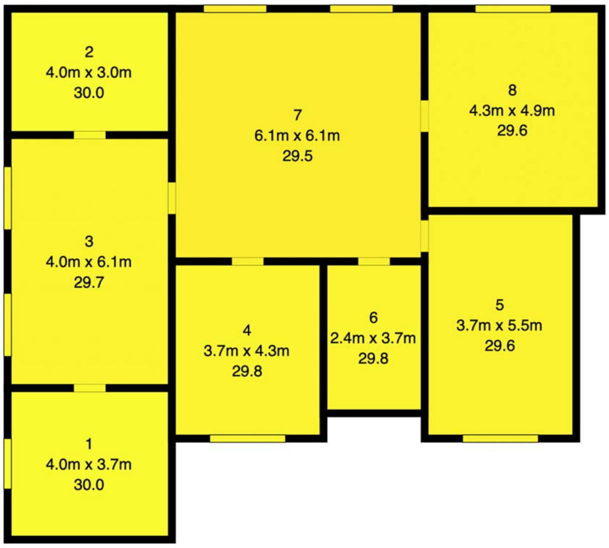

Fig. 1 Medium-size house layout

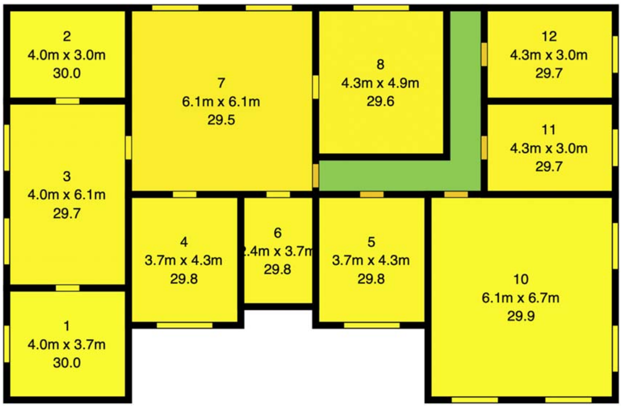

Fig. 2 Small-size house layout

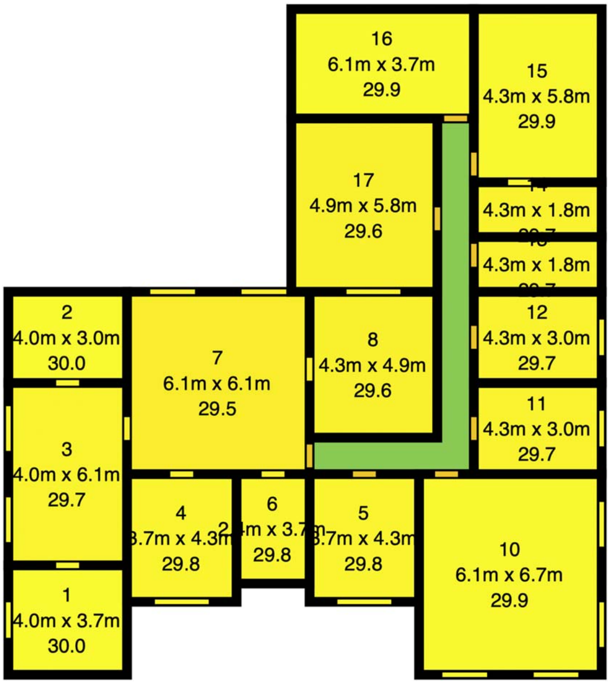

Fig. 3 Large-size house layout

3 Investigating Strategy Transfer in CISAT

This section investigates how design strategies can be transferred across problems. In order to transfer learned strategies, the first step is efficient modeling and representation of those strategies. Sec. 3.1 illustrates how HMMs may be used as an adaptive operation selection model with features that can capture the state-dependent operation sequencing. Sec. 3.2 enlists and evaluates the various strategies that have been extracted from the human designer data. Finally, in Sec. 3.3, the relative performance of strategies is compared over different problems, and final results for strategy transfer are discussed. 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 an Adaptive Operation Selection Framework. In the CISAT framework, MC models with online learning were efficient in learning design sequences and helped the agents perform similarly to the human designer. Compared with MC models, HMMs provide more structured information about the design process [10]. However, to the authors' knowledge, HMMs have not been used to guide operation selection for computational agents. This work (1) demonstrates the transfer of strategies from humans to computational agents using HMMs and (2) demonstrates the transferability of these HMM-encoded strategies across problems.

The state-based preferences encoded in an HMM portray a behavior similar to how designers have different strategies during sequential phases of design, showing how designers adapt their process according to emerging situations [28]. Two different approaches are used to learn the parameters of the HMM: offline and online learning. Baum–Welch algorithm is [29] used to learn the model parameters from a dataset of operation sequences. Model parameters are determined offline, creating a model that captures certain aspects of the human design behavior present in the historical data. Online learning, on the other hand, means real-time learning that occurs during the process of solving the design problem. For a human designer, this would mean attempting different operations and then updating beliefs by observing positive results. The agents in CISAT learn online using update rules based on a learning heuristic identified by Simon and Kotovsky [30], which updates and improves the strategies by reinforcing the probabilities that led to an increase in performance. This update approach is also similar to pursuit approaches used in multi-arm bandit problems [31] and reinforcement learning [32]. This is an important feature of CISAT that makes the agents adaptive. This update occurs at every iteration, which is analogous to discrete time steps, a set of natural numbers.

HMMs have two types of model parameters: operation emission and state transition matrix. For the problem in this study, there are nine operations and four hidden states (representing internal states of the agent, determined previously in Ref. [10]). The transition matrix T is a 4×4 matrix, representing probabilities of state transitions. The emission matrix, E, stores the weights for selecting an operation given a particular state (4×9 matrix). Throughout the design process, it is these two matrices (44 degrees of freedom) that govern the operation selection. The weights of the matrices along with the online learning process determine the strategy that the agents employ to solve the design process and for the same reason have been referred to as design heuristics previously [10]. T_{ij} refers to the probability of transitioning from state i to state j. Similarly, for the emission matrix, E_{ik} refers to the probability of choosing a particular operation k given a state i. At every iteration, the agent stochastically chooses the next state from transition matrix T and then stochastically chooses the operation based on the new state from E_{ik}. A higher value of the respective element makes that state or operation more preferable. The online learning method that is employed for the HMM is similar to that used previously by McComb et al. for MCs [9]. It works on collecting real-time reward based on the change in the quality and then proportionally reinforces the probability weights that led to that reward. So, if state s leads to a state s' and an operation a is selected, which causes a change in the quality of the solution, and then, the reward updates the weights by the operations shown in Eqs. (1)–(4):

$$T_{ss'} \leftarrow T_{ss'} \cdot (1 + \Delta \cdot lr) \tag{1}$$

$$T_{ss'} \leftarrow \text{Normalize}(T_{ss'}) \tag{2}$$

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

$$E_{s'a} \leftarrow \text{Normalize}(E_{s'a}) \tag{4}$$

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 and positive. The variable lr is the learning rate, which determines how greedily the probability is changed in the process. This online learning method is based on the method previously utilized in CISAT [26] and has been adapted here to be used with HMMs to maintain uniformity when comparing different operation selection models in CISAT. Standard algorithms like Baum–Welch [29] are not stable for online learning in their original formulation [33], and significant advancements [33,34] need to be implemented for application in an online setting, an area to be explored in future work.

3.2 Using Diverse Design Strategies to Generate Designs. The parameter priors 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 section explores the effects of having different parameter priors on the behavior of computational agents. All the agents in a team follow identical initial strategies (priors) and hence have the same initial probability values for the HMM matrices. Five sets of different strategy priors were used for comparison. Figure 4 shows the values of the different strategy priors leading to different design strategies. It must be noted that the values represented in Fig. 4 are only the initial values for these strategies and act as priors, which 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. Apart from the different strategies, all the team characteristics (agent interaction, self-bias, quality-bias among others mentioned in Ref. [9]) were kept constant for the experiment, and the simulation was repeated for 40 trials. The experiment was conducted on a medium-size house layout, which was also the original design problem used in the study to extract designer strategies; hence, the results evaluate the performance of transferring the strategies from humans to agents. The experiment was terminated at 50 iterations to mirror the original problem in the design study. Because it is a time-constrained problem, the strategies required to solve the problem might vary if the number of iterations were increased. Thus, to maintain similarity and a fair comparison with the timing of human solvers, the agents were only allowed 50 iterations to design and learn. The description of these strategies is as follows:

(1) *Best and Worst Problem Specific*: The weight distributions in this set are learnt offline from a previous human design study [10] on the same cooling problem as mentioned in Sec. 2.3. The Best Problem Specific strategyrepresents the strategies that high-performing designers used in the design study, and similarly, for the Worst Problem Specific strategy show worst-performing designer strategies. These have been named Problem Specific since these are exact values from the design study on a particular problem.

(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. These matrices were created by initializing the matrix to small non-zero values. Then higher weights were added based on highly weighted elements in the Problem Specific strategy. Finally, the values were normalized to convert into probabilities. This allows all elements to have non-zero weights, which makes it different from the Problem Specific set, increasing the randomness in the model allowing the agent to explore even the nonpreferred operations.

(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.

The results in Fig. 5 show the mean performance of the different strategies along with standard error bars and 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 the designer behavior. Second, since there is a significant gap between the random initialization strategy and the human-derived inputs (both heuristic and problem specific); the importance of using design knowledge (probability priors) to perform better is illustrated. This is an example where

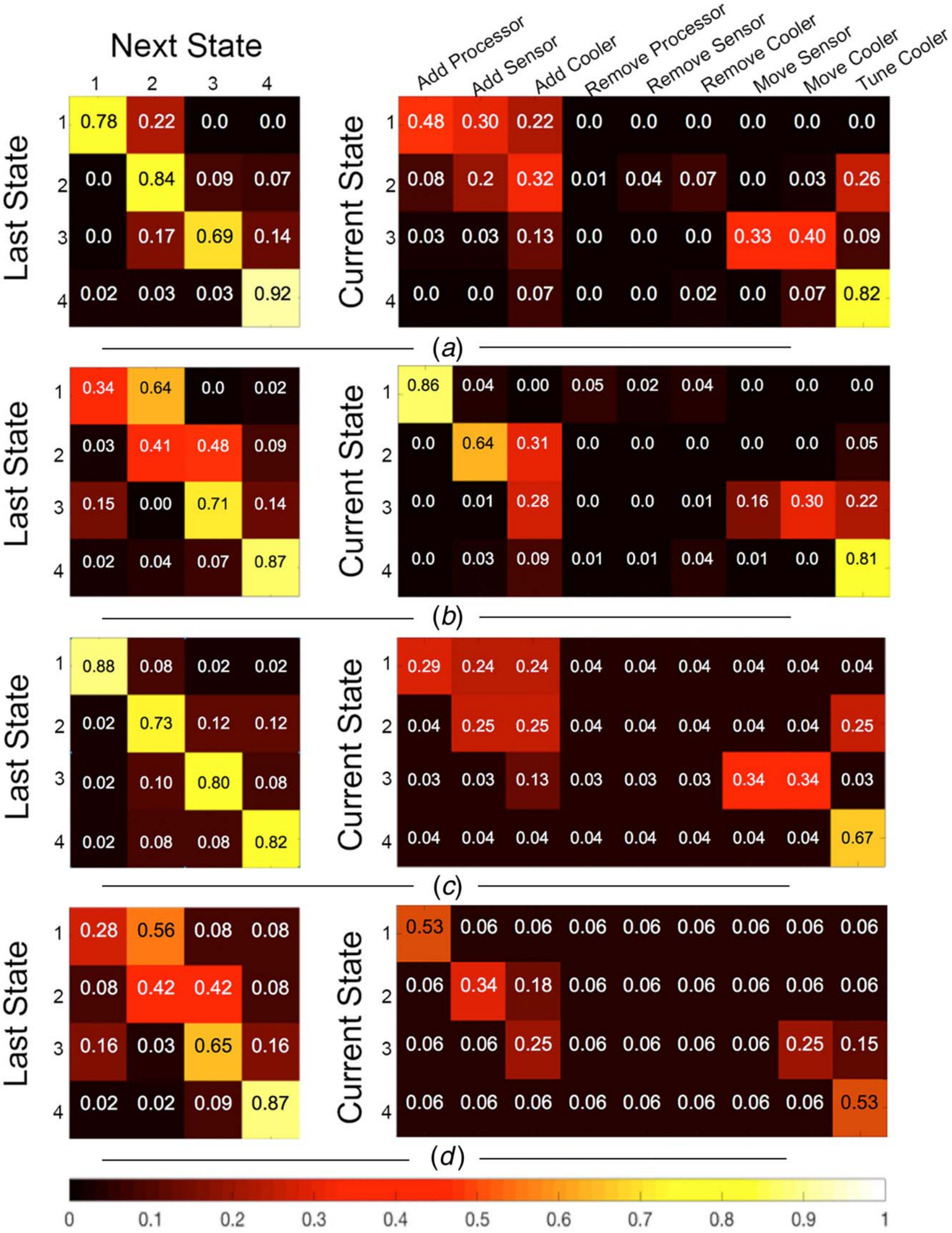

Fig. 4 State transition matrices (left) and emission matrices (right) for following strategy priors: (*a*) Best Problem Specific, (*b*) Worst Problem Specific, (*c*) Best Heuristic, and (*d*) Worst Heuristic. The gradient bar indicates the probability of selection.

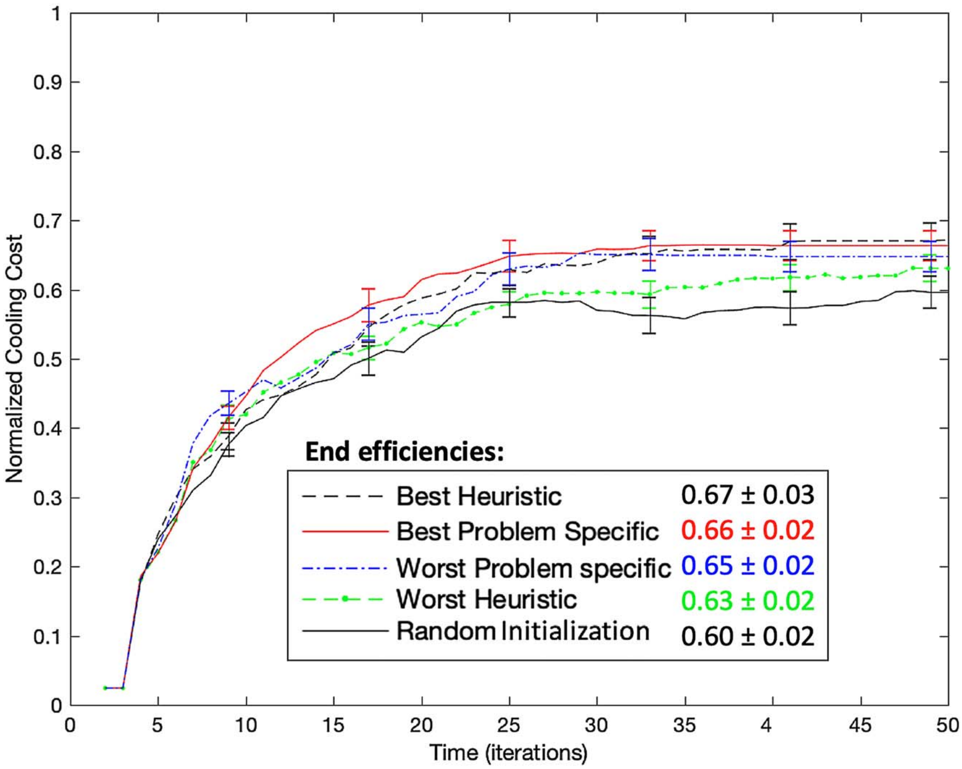

Fig. 5 Design strategy comparison for medium-size house layouts. The error bar represents ±1 standard error.

strategies were successfully transferred from humans to machines (agents) through the use of these probabilistic computational models.

3.3 Design Strategy Transfer. This section tests the performance of the CISAT teams when the problem being solved is different from (but the same type as) the problem on which the strategies were learned. This investigation has goals similar to the concept of transfer learning in reinforcement learning [35]. The goal of transfer learning is to obtain pretrained values (or strategies) and to use them for a new problem. Torrey and Shavlik [36] 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 section presents computational experiments that compare the different strategies transferred from the medium house and uses certain measures [36] to compare their performance for the successful transfer of design 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. The problems have the same operations and states for solving them.

For the small-size house, the results are shown in Fig. 6. This problem is computationally less intensive since this layout has fewer rooms, which is shown by the different strategies leading to similar final efficiencies. However, a faster increase and higher final value can be observed for strategies derived from the human data (both problem specific and heuristic) compared with random initialization. The random initialization strategy shows comparatively poor efficiency values in the beginning but a high final efficiency; this is due to online learning improving the initial bad strategy over time. This effect shows that the online learning can enhance a bad strategy even over a limited number of iterations, motivating future work to test the effectiveness of different online learning algorithms. The Best Heuristic and Best Problem Specific strategies are much higher in the beginning in comparison with Random, showing that transferring human strategies help achieve better solutions faster in the process. The large house is a more complicated design problem than both the small and the medium house; performance results for it (shown in Fig. 7) show that the best heuristics and best problem specific strategies perform considerably better than the rest. Overall the best performance was achieved by Best Heuristics, showing that a generalized human design strategy can perform equal to or better than problem specific strategies when transferred to new unseen problems. The Random Initialization strategy performs poorly showing that online learning itself is not able to learn good strategies on a complex problem. Both the results in Figs. 6 and 7 show that 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. These observations provide a successful example of transfer learning in accordance with Torrey and Shavlik [36]. The Best Heuristics worked the best for all the three different sizes of the house. Having non-zero probabilities for seemingly nonoptimal operations allows the model to explore new operations. 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.

4 Conclusion and Further Discussion

This work explores open questions relevant to agent-based computational design and strategy transfer in design. Specifically, the note explores the representation of design strategies as an HMM,

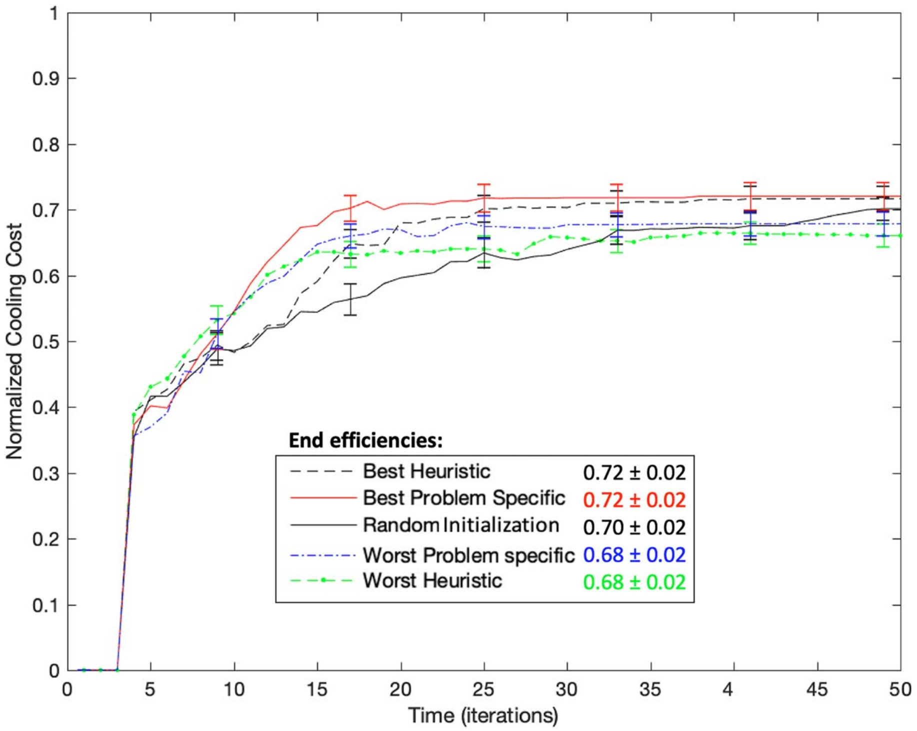

Fig. 6 Design strategy comparison for small-size house layouts. The error bar represents ±1 standard error.

their application to engineering design problems, evaluation of their generality, and transferability across similar problems. Broadly speaking, all the findings of this paper can be summarized in two major results.

First, the work demonstrates the successful transfer of design strategies from human designers to computational agents. The observations made in Sec. 3.2 show that design strategies from human designer data performed similar trends even when applied by computational agents. Human design heuristics were successfully represented through probabilistic models and hence establish a common ground between human designers and computational agents for representing design strategies. This opens up new

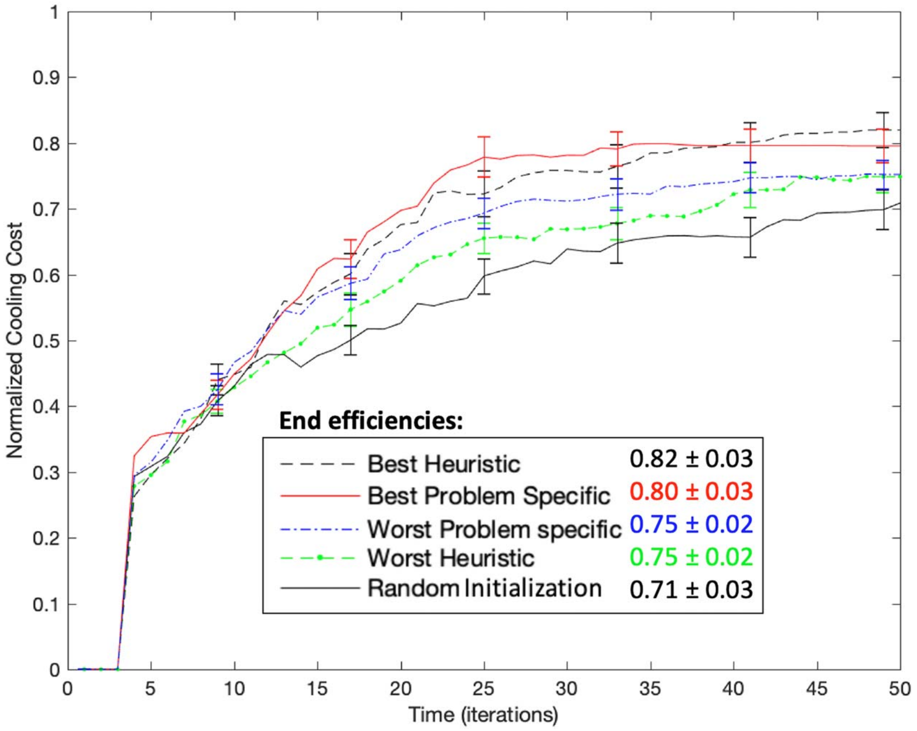

Fig. 7 Design strategy comparison for large-size house layout. The error bar represents ±1 standard error.

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 systems across similar but new problems. These models leverage past human experiences using offline learning to augment the abilities of computational agents. The results and experiments conducted in Sec. 3.3 begin to explore this intersection of transfer learning and engineering design. The strategies are tested on three different problems, and the Best Heuristic strategy learned from human data shows the best performance. An increase in agent performance is seen on different problems especially in the beginning of the design process, and this signifies that using previous experience in the form of design strategies helped agents perform better and faster than a randomly initialized agent. These inferences illustrate how previous design experiences could be used to develop generalizable strategies and improve performance across new design tasks.

The work illustrates the importance of extracting human strategies from the data for generalized learning across problems. However, the approach has several implicit limitations. The quality of the extracted strategies is dependent on the performance or skills of the human subjects; however, this can be avoided if strategies are extracted from the data provided only by high performing humans. Also, the current work deals with transfer over problems with state/operation spaces, which are similar and relatively small in size; future work can explore the effectiveness of the methodology on different or larger state/operation spaces.

Acknowledgment

An extended version of this paper was presented at *IDETC'18* [37]. This material is based on work supported by the Defense Advanced Research Projects Agency through cooperative agreement No. N66001-17-1-4064 (Funder ID: 10.13039/100000185). 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] Perkins, D. N., Salomon, G., and Press, P., 1992, "Transfer of Learning," International Encyclopedia of Education, 2nd ed., Pergamon Press, New York.

[2] Hoffman, R. R., 1998, "How Can Expertise Be Defined? Implications of Research From Cognitive Psychology," *Exploring Expertise: Issues and Perspectives*, R. Williams, W. Faulkner, and J. Fleck, eds., Palgrave Macmillan UK, London, pp. 81–100.

[3] Qureshi, M. O., and Syed, R. S., 2014, "The Impact of Robotics on Employment and Motivation of Employees in the Service Sector, With Special Reference to Health Care," Saf. Health Work, **5**(4), pp. 198–202.

[4] Mittal, S., and Frayman, F., 1989, "Towards a Generic Model of Configuraton Tasks," Proceedings of the 11th International Joint Conference on Artificial Intelligence—Volume 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 1395–1401.

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

[6] Maher, M. L., 1990, "Process Models for Design Synthesis," AI Mag., **11**(4), p. 49.

[7] Breuker, V. D. V., 1994, *Common KADS Library for Expertise Modelling*, IOS Press, Amsterdam, The Netherlands.

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

[9] 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.

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

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

[12] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Capturing Human Sequence-Learning Abilities in Configuration Design Tasks Through Markov Chains," ASME J. Mech. Des., **139**(9), p. 091101.

[13] Ji, H., Yang, M. C., and Honda, T., 2007, "A Probabilistic Approach for Extracting Design Preferences From Design Team Discussion," ASME. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 3: 19th International Conference on Design Theory and Methodology; 1st International Conference on Micro- and Nanosystems; and 9th International Conference on Advanced Vehicle Tire Technologies, Parts A and B, Las Vegas, NV, Sept. 4–7, pp. 297–306.

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

[15] Olson, J., Cagan, J., and Kotovsky, K., 2009, "Unlocking Organizational Potential: A Computational Platform for Investigating Structural Interdependence in Design," ASME J. Mech. Des., **131**(3), p. 031001.

[16] 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., New York.

[17] 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.

[18] 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.

[19] 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.

[20] Perišić, M. M., Martinec, T., Štorga, M., and Kanduč, T., 2016, "Agent-Based Simulation Framework to Support Management of Teams Performing Development Activities," 14th International Design Conference (DESIGN 2016), Dubrovnik, Hrvatska, May 16–19, pp. 1925–1936.

[21] 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, Vancouver, Canada, Aug. 21–25.

[22] 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., **11**(3), pp. 172–192.

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

[24] Miller, S. W., Simpson, T. W., and Yukish, M. A., 2015, *Design as a Sequential Decision Process: A Method for Reducing Design Set Space Using Models to Bound Objectives*.

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

[26] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks," *Des. Comput. Cognit.*, Vol. 16, J. Gero, ed., Springer, Cham, pp. 401–418.

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

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

[29] 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. Statist, **41**(1), pp. 164–171.

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

[31] Kuleshov, V., and Precup, D., 2014, "Algorithms for Multi-Armed Bandit Problems," arXiv preprint.

[32] Sutton, R. S., and Barto, A. G., 1998, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA.

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

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

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

[36] Torrey, L., and Shavlik, J., 2009, "Transfer Learning," *Handbook of Research on Machine Learning Applications*, E Soria, J Martin, R Magdalena, M Martinez, and A Serrano, eds., IGI Global, Hershey, PA, pp. 1–22.

[37] Raina, A., McComb, C., and Cagan, J., 2018, "Design Strategy Transfer in Cognitively-Inspired Agents," ASME. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 2A: 44th Design Automation Conference, Quebec City, Quebec, Canada, Aug. 26–29.