**Proceedings of the ASME 2015 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2015
August 2-5, 2015, Boston, Massachusetts, USA**

DETC2015-47674

PARAMETER DESIGN STRATEGIES: A COMPARISON BETWEEN HUMAN DESIGNERS AND THE SIMULATED ANNEALING ALGORITHM

Bo Yang Yu *
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Olivier de Weck
Department of Aeronautics and Astronautics
and Engineering Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

Maria C. Yang
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

Computer-based tools have great potential for facilitating the design of large-scale engineering systems. Interviews with veteran designers of desalination systems revealed that they tended to employ a trial-and-error approach to determine critical design parameters when using software design packages. A series of human experiments were conducted to observe the performance and behavior of test subjects during a series of simulated design processes involving seawater reverse osmosis (SWRO) plants. The subjects were mostly students with a spectrum of knowledge levels in desalination system design. The experiments showed that subjects who ranked top in performance behaved very differently from those who were bottom-ranked. The problem-solving profiles of the best performing subjects resembled a well-tuned simulated annealing optimization algorithm while the worst performing subjects used a pseudo random search strategy. This finding could be used to improve computer-based design tools by utilizing the synergy between strengths of humans and computers.

————————

*Address all correspondence to this author, byyu@mit.edu.

INTRODUCTION

The design of complex engineering systems requires designers to manage a very large number of parameters across disciplines and subsystems, balance the trade-offs between cost, performance, reliability or other measures, while satisfying an array of conflicting design requirements and constraints. With advancements in computing, numerical simulation, and meta-modeling techniques, computational design tools can now analyze thousands of design alternatives cheaply and quickly [1, 2]. While advancements in design automation have given human designers many powerful tools, it has also placed great responsibility on the designer to make sound decisions with the design tool. Researchers have realized the importance of the interaction between designers and numerical design tools, and have shifted toward including the designers "back in the loop" in design automation research [3, 4]. Due to their importance in design, understanding the behaviors and needs of the human designers is critical to finding potential areas of improvement in the design process.

To understand the current practice of engineering design at the designer level, the authors conducted a series of four interviews with practitioners working in the desalination and water

1

production industry. The interviews revealed that the desalination industry relies on the expertise of human designers to evaluate different design alternatives with the aid of design evaluation software. Design parameters are tweaked in an empirical fashion until system level requirements are met [5].

This design process described by the practitioners can be classified as a parameter design process [6]. A parameter design problem is usually well defined, with clearly identified design variables (the input parameters), objectives and constraints (the output parameters). Past research in the area of parameter design problems have shown that, due to cognitive limitations, human designers are very inefficient at solving generic parameter design problems using computational tools [6–8].

Numerical optimization methods such as gradient descent, genetic algorithms, and simulated annealing can be used to solve parameter design problems, by taking advantage of the growing computational capabilities of modern computers. However, the interviews revealed that software design tools in industry do not have optimization capabilities. This is because optimization tools for desalination are mostly academic, and extensive customization is required for real-world industrial applications. Broadly speaking, optimization-based design tools have seen limited use in many industries due to reasons that include: 1) lack of maturity of optimization algorithms [3], and 2) uncertainties in design preference formulation [9, 10]. Because of the inherent disadvantages of design automation, human designers are tasked with solving parameter design problems in most industry engineering design practice.

In this paper, human designer behavior during parameter design is investigated through a series of laboratory experiments. Test subjects are recruited from academia are asked to complete a set of constraint-satisfaction design tasks for complex system design, in this case a reverse osmosis desalination system. Then, the process that human designers used to search for satisficing designs are analyzed. The following two questions will be answered in the scope of this paper:

1. Does differences in strategies (such as the values of design variables changed at each iteration) lead to differences in test subjects' performances?
2. Are there similarities between designers' strategies and existing numerical optimization algorithms?

These research questions provide a human-centered approach to engineering design research, and propose to model the socio-technical process of parameter design using a physical process of numerical optimization. By understanding the behaviors of human designers, future improvements to software design tools could be made to enhance the designer's efficiency in parameter design tasks.

BACKGROUND

Parameter Design Problems

Parameter Design Problems are typically associated with the detailed design stage in engineering system design, where the designer manipulates a set of input variables (design parameters) to make changes in a set of output variables (functional requirements or performance parameters) [6].

Studies have been conducted in the past decade related to humans' abilities to solve computer-based parameter design problems. Hirschi and Frey were one of the first to conduct parameter design experiments on human subjects [6]. They performed a series of experiments in which human subjects were asked to solve generic mathematical parameter design problems that ranged from 2-input-2-output (2x2) parameters to 5-input-5-output (5x5) parameters, using a custom-built computer user interface. They found that the time taken to solve parameter design problems was on the order of $O(n^{3.4})$, where n is the number of input parameters, when the input parameters are coupled (changing any one input parameters affect multiple output parameters). Grogan compared solving parameter design tasks individually vs collaboratively, using test problems similar to Hirschi and Frey's experiment. His results were in agreement with the results of Hirschi and Frey for both individual and collaborative tasks [8]. Flager, et al. conducted similar research using parameter design problems specific to the building design domain [7]. Their results indicated that the design solution quality found by their test subjects decreased with an increase in problem scale, following a power-law relationship similar to that reported by Hirschi and Frey. Austin-Breneman, et al. conducted a set of collaborative design optimization experiments where human subjects were asked to optimize a simplified satellite design system, and these experiments illustrated humans can be inefficient at optimization with little system level awareness [11]. Ligetti and Simpson published a series of studies on user performance with software design interfaces, and they found that the number of problem variables and computational delays negatively affected user performances [12, 13], while the "richness" of the design interface increased user performance [14].

All of these studies suggest that human designers are not efficient at solving coupled parameter design problems. Human designers' performance on parameter design problems strongly depend on the number of variables in the problem, the technical context of the problem, and the user-friendliness of the software interface. Past studies have discussed the differences in strategies that designers use, but no study has systematically examined the nature of these strategies in detail. In this work, a set of human experiments are conducted to specifically investigate the human behaviors while solving parameter based design problems, and makes active comparisons between human strategies and numerical optimization algorithms.

2   Copyright © 2015 by ASME

METHODOLOGY

A set of experiments were conducted in which human subjects were asked to complete a series of design tasks in the context of reverse osmosis desalination. The subject's performance was observed, recorded and analyzed to reveal for strategies and patterns.

Experimental Procedure

22 test subjects participated in this study including 2 undergraduate students, 3 masters students, 14 PhD students, and 3 post-docs all affiliated with MIT. All subjects had experiences in mechanical design. 14 of the 22 subjects had varying educational or work-related experience with designing desalination systems. Only 5 of the subjects had experiences with numerical optimization methods.

Participants were sat at a PC running the experiment software interface. The subjects were provided informed consent, and filled out a background questionnaire that asked questions about their knowledge and experiences in designing desalination systems. Next, a short introduction of the design tasks was given to the subjects, and the subjects would then perform the design tasks uninterrupted. A pen, paper and a calculator were provided as optional external aids. An experimenter were present to make notes of any observations. Subjects were also interviewed at the end of the experiment.

Seawater Reverse Osmosis Design Tasks

Participants were asked to determine the flow and equipment properties of a 2-pass seawater reverse osmosis (SWRO) desalination plant to satisfy a set of design constraints. The 2-pass configuration is common for a desalination plant, and would be familiar to those with experiences in desalination. A numerical simulation of the reverse osmosis process was developed in MATLAB based on the solution diffusion model [15–17]. There were ten input variables and five output variables in total as listed in Table 1. In addition, there were 8 physical constraints on membrane performance provided by the membrane supplier. A detailed discussion of the model can be found in [5].

To maintain a level of consistency with previously published literature, the numbers of input and output parameters were varied to create five different design tasks: 2x2 (2 inputs and 2 outputs), 3x3, 4x4, 5x5, 10x5. Each output parameter was assigned a constraint value which the test subject must achieve. Any unused parameter would be assigned a very generous target that would always be met, and any unused input parameters would be kept constant. The performance target values for each problem were diversified to reduce learning effect between problems, while at the same time keeping the numerical complexity of all problems consistent. All test subjects were presented with the same five design tasks, but in pseudo-random order.
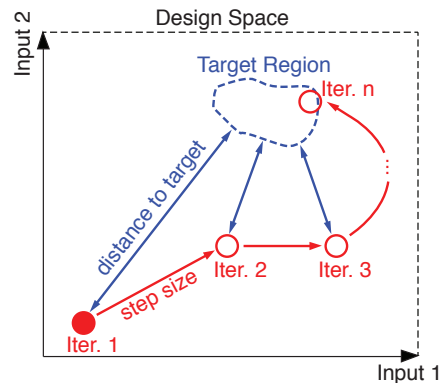

FIGURE 2. Design space, target region, and problem solving process.

Software Interface

The design problem was presented to the test subjects in a custom software interface that was built for the purpose of the experiment. This software had fewer features compared with a similar commercially available design evaluation software, so that designers with different levels of desalination knowledge could be tested, while simplifying the data collection process.

The user interface software was created in MATLAB to interact with the RO simulation. The interface took some design cues from the Reverse Osmosis System Analysis (ROSA) software distributed by Dow Chemical. A screenshot of the user interface is shown in Figure 1. All input parameters were listed in the top-left panel, all output parameters and constraints were listed in the bottom-right panel, and any unsatisfied constraints were highlighted in red. The top-right panel had an image of the 2-pass SWRO process and intermediate variables.

At the beginning of each design task, all input variables were set to their default values at the lower bound of their range. The users would manipulate the sliders and drop-down menus to change the input variable values, but the software only evaluated the design and update the output parameters when the "calculate" button was pressed. Each click of the "calculate" button was considered as one design *iteration*. The computation time for each iteration was less than one second. The users would continue to modify the input variables and evaluate their designs to satisfy constraints. Once all constraints were satisfied, the GUI automatically informed the user that they have competed the current design task and to proceed to the next design task. The software interface automatically collected input and output parameter values, number of constraint violations, and time elapsed at each iteration.

Performance Metrics

To understand the strategies subjects uses, several metrics were defined and calculated. Figure 2 is a notional figure of the design space and the search process of the participant. The blue

TABLE 1. List of variables in the design experiment problem

| Input Variables	Range	Output Variables	Constraint
1. feed flow rate*, q_{in}	400 - 1000	1. product flow rate, q_p	$q_{p,min} \leq q_p \leq q_{p,max}$
2. permeate blending flow rate, q_{ff}	0 - 2.5	2. product TDS, c_p	$c_p \leq c_{p,max}$
3. recovery, pass 1, r_1	30% - 60%	3. product boron content, b_p	$b_p \leq b_{p,max}$
4. recovery, pass 2, r_2	40% - 90%	4. energy consumption, E_c	$E_c \leq E_{c,max}$
5. # of pressure vessels, pass 1, npv_1	1 - 200	5. capital cost, CC	$CC \leq CC_{max}$
6. # of pressure vessels, pass 2, npv_2	1 - 50		
7. membranes / vessel, pass 1, n_{memb1}	1 - 8		
8. permeate split, psplit	1 - 8		
9. membranes / vessel, pass 2, n_{memb2}	1 - 8		
10. brine recirculation ratio, r_{bb}	0 - 100%		

* flow rate measured in m^3/h
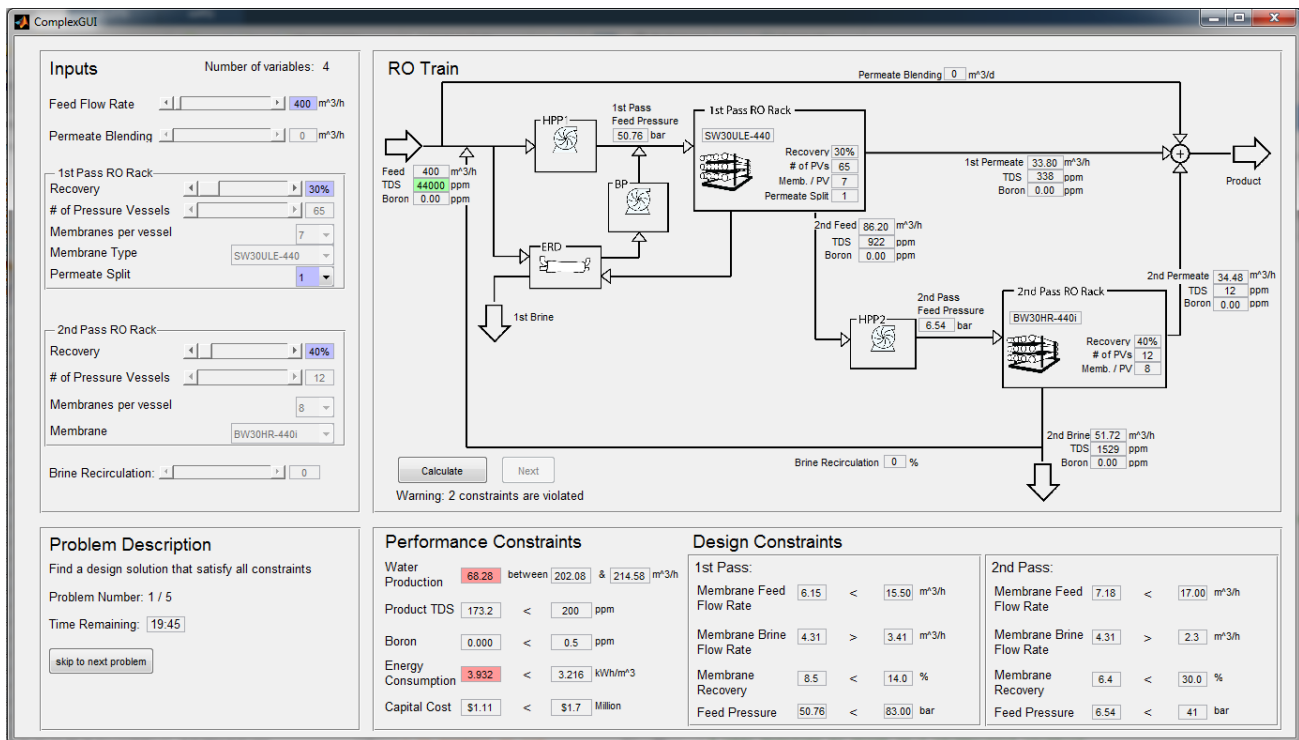
FIGURE 1. Graphical software user interface for the test problem

dashed-line represents the target region containing all possible designs that satisfy all constraints. The solid red circle shows the initial design point, which is always at the lower limits of input parameters, and the hollow red circles represent the design points at each iteration. Three detailed metrics are shown: Distance-to-target, step-size, and number of parameters changed each iteration.

The distance-to-target metric is defined as the shortest distance in the design space from the current design point to the target region. The target region for each test problem was found through latin-hypercube sampling of the design space, plus the solutions found by all test subjects during experiment.

The step-size is the distance between designs from consecutive iterations. Distance metrics were computed *post facto* based

4

Copyright © 2015 by ASME

on the experiment log. The Manhattan distance was selected over others such as Euclidean distance because most subjects only changed one variable at each iteration, and thus the Manhattan distance was most similar to how the participants navigated the design space in this particular experiment. Input variables were normalized to values between 0 and 1.

RESULTS

Results of the experiments are discussed in this section, including an overview of the data, and detailed comparison between test subjects' strategies and the simulated annealing algorithm.

Overview

Not all test subjects were able to complete all design tasks during the required time period. On average, a subject completed 4.3 design tasks out of the 5 tasks provided, for a completion rate of 86%. Therefore the time and iterations measurements collected would be right-censored[1] in nature. For all statistical analyses in this study that involved censored data, the log-rank test based on the Kaplan-Meier procedure was used to compare significant differences between two populations [19]. A modified Spearman's correlation test was used to find correlation coefficients involving censored measurements, where any censored measurements would be multiplied by a constant factor of 2 before being used in the correlation analyses [20]. Comparison of uncensored data were done using Wilcoxon rank sum test, which is a non-parametric test that does not assume a prior distribution of data points.

To check whether the order of problems presented to the test subjects affected their performance, log-rank tests were used to compare the average performances of problems solved in different order. Both the learning effect and fatigue effect were checked. The learning effect was defined as performance increase (reduced completion time and iterations) from the first task to the second and third tasks, and fatigue effect was defined as the performance decrease from the second and third tasks to the fourth and fifth tasks. No significant ordering effects were found except for the 10x5 problem, which showed that subjects perform better (shorter time and lower iterations) when given the problem last.

All subjects in the experiment spent roughly the same amount of time on each design iteration. All analyses in this work were performed with both time and iterations measurements and produced the same conclusions. Only the analyses using the iterations measurements are reported in this paper.

A "performance ranking" was calculated for each test subject using the following method. First, subjects were ranked for
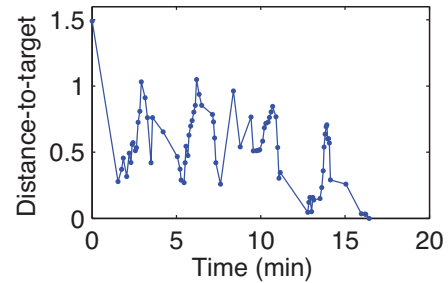
---

[1] duration measurements that terminated prematurely [18]

FIGURE 3. Example plot of distance to target data

each design task based on the iterations taken to solve them. Subjects who were tied were assigned equal rankings, and subjects who did not complete a problem were considered to be tied for last. Then, a weighted average of the rankings of all five design tasks were calculated, and the subjects are ranked again based on the average to determine the subjects' performance rankings. The weights are proportional to the median iterations taken to solve each problem, so that the rankings of more difficult problems were given a higher weight. Subjects with top performance rankings solved the design tasks in shorter time and fewer iterations in general.

Subjects with strong desalination knowledge generally ranked higher in performance. Based on the recordings of the experiments and interviews, there seem to be four different classes of strategies taken by the test subjects: 1) subjects' prior knowledge of the system matched the design problem and they were able to apply their knowledge. 2) subjects' prior knowledge did not match the design problem. 3) subjects had no prior knowledge but were able to learn the relationships between parameters. 4) subjects had no prior knowledge and could not learn the relationships between parameters. These observations are discussed in detail in [5]. The rest of the results section focus on the characterization of the parameter design process, the relationship between the characteristics and performance ranking.

Strategy Comparison to Simulated Annealing

Figure 3 shows the distance-to-target plot of a test subject trying to complete a design task. It is immediately evident that this subject consistently made his/her designs worse over multiple iterations, shown by the distance-to-target value increasing in multiple occasions. This phenomenon of going away from target was observed in 17 of the 22 subjects, and lead the authors to hypothesize that the heuristic optimization algorithm simulated annealing may have similarities to the problem solving strategies that was used by the subjects: the random uphill-move in simulated annealing is analogous to the phenomenon of going away from target observed during the experiments.

Simulated annealing searches for an optimal design (minimizing an objective function) by iteratively exploring the de-

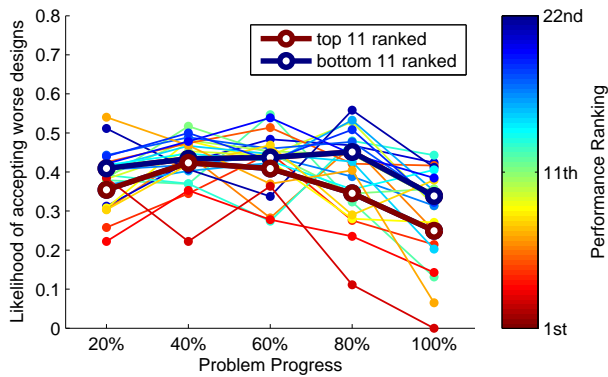5   Copyright © 2015 by ASME

FIGURE 4. Comparison to simulated annealing: subject's temperature profile. Estimated by the likelihood of accepting worse designs.

TABLE 2. Correlation between likelihood of accepting worse design at each progress interval and performance. Positive correlation suggest that higher performance ranking correlates to lower likelihood of accepting worse designs

Problem Progress	20%	40%	60%	80%	100%
Spearman's ρ	0.47	0.29	0.21	0.67	0.52
p-value	0.03	0.19	0.34	0	0.01

TABLE 3. Correlation between step-size at each progress interval and performance. Negative correlation suggest that higher performance ranking correlates to bigger step-size

Problem Progress	20%	40%	60%	80%	100%
Spearman's ρ	-0.74	-0.53	-0.50	-0.33	-0.29
p-value	0	0.01	0.02	0.13	0.2

sign space. The algorithm is inspired by the annealing process for physical materials like metal and polymers, where slow cooling (annealing) allows the material to arrive at the lowest energy state. Inherent random fluctuations in energy allows the annealing system to escape local energy minima to achieve the global minimum.

At each iteration, a trial design is randomly selected in the neighborhood of the current design. The algorithm moves to the trial design if it is better (lower objective function value) compared to the current design. If however, the trial design is worse compared to the current design, the algorithm still accept trial design randomly, with a probability that is proportional to the "temperature" of the algorithm. Temperature of the algorithm is slowly decreased following the "cooling schedule" of the simulated annealing algorithm. A decreasing temperature profile (cooling schedule) ensures that the design space is explored randomly at first, and then focuses on improving the design toward the end of the search. Details of simulated annealing can be found in Appendix A.

Cooling Schedule To compare whether the test subjects' search patterns exhibit any characteristics of a simulated annealing algorithm, their "cooling schedules" were estimated based on their likelihood of going away from target (accepting a worse design). Each subject's progress through the design problems was equally divided into five 20%-segments based on the percentage of overall iterations. The ratios of iterations subjects spent going away from the target to all iterations was computed for each segment, which is an indication of the likelihood that the subject was headed to a worse design at each iteration, and can be considered as the "temperature profile" of the subject. The "true temperature profile" of the test cannot be exactly computed since the probability of accepting a worse design at each iteration cannot be computed exactly.

Figure 4 shows this likelihood through the problem progress.

Each line represents a test subject, and the lines are color-coded according to the subject's performance ranking. The average values for the top 11 and bottom 11 ranked subjects are also plotted for ease of comparison.

Figure 4 shows that for faster subjects (on the red end of the color map), the likelihood of accepting a worse design drops toward zero as they progress through the problem, whereas the slower subjects (blue end of the color map) have likelihood values that hover around 0.4 throughout the problem. This is in good agreement with the simulated annealing algorithm, the "temperature" must decrease over time for the algorithm to converge. Table 2 shows correlation analysis between the likelihood of accepting a worse design and the subject's performance ranking at each progress interval. Statistical significance was found in the last 40% of problem progress, which confirms the observation: subjects who are faster have decreasing "likelihood of accepting worse design" that is analogous to the cooling schedule of simulated annealing.

Neighborhood Size The size of the "neighborhood" in simulated annealing is analogous to the step-sizes subjects took at each iteration. A common practice in simulated annealing uses the variable neighborhood approach, where the neighborhood size is varied over each iteration, proportional to the temperature [21].

Figure 5 shows the average step-size change over the progress of a problem for each test subject. All subjects start with a large step-size in the beginning of problem and decrease as the problem progresses, similar to the variable neighborhood approach used in simulated annealing. The faster subjects tend to have slightly larger step-sizes, but the shape of the plots look
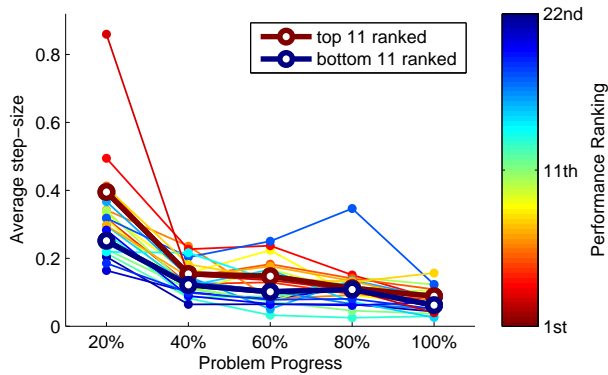
6   Copyright © 2015 by ASME

FIGURE 5. Comparison to simulated annealing: subject's search neighborhood. Estimated by average step-size.

largely the same for all subjects. Table 3 shows the correlation analysis between step-size at each progress interval and performance ranking. The results show that larger step-sizes in the early stages of the design process tend to have a stronger correlation to subject's performance.

Performance Comparison to Simulated Annealing

A simulated annealing algorithm is set up to solve the design tasks used in the experiment to be compared with the performance of the test subjects. The objective function of the algorithm is defined as the sum of the absolute margins of unsatisfied constraints. The algorithm was tuned to stop automatically when the objective function reaches 0, or after 600 iterations. The annealing temperature is assumed to start at 100 and decrease exponentially with a multiplication factor of 0.9 at each iteration. The search algorithm was run a hundred times with the neighborhood size varying from 0.01 to 0.4. Only the 3x3 design task is presented in this paper because the number of iterations taken by the simulated annealing algorithm was on the same order compared to the test subjects, and simplifying the presentation of data. The total number of iterations vs mean step-size for the simulated annealing algorithm is shown in Figure 6 as red dots.

A third degree polynomial is fitted to the simulated annealing results (shown in Figure 6 solid black line). The trend suggests that for very small step-sizes, more iterations are required for simulated annealing, which is consistent with the experimental observation. The results also show that if the step-size is too high (above 0.3) there appears to be an increase in the number of iterations required, since too large of a neighborhood size makes the simulated annealing algorithm behave more like a random search [22].

The iterations each test subject took to complete the 3x3 problem are plotted in Figure 6 as green circles. Human test subjects tended to take fewer iterations compared to the simulated annealing algorithm, which is to be expected, since simu-
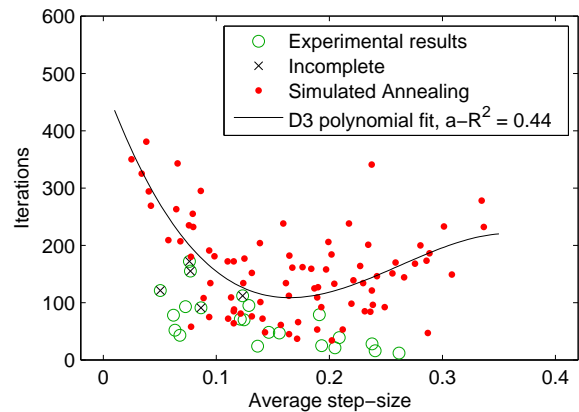

FIGURE 6. Iterations vs average step-size, comparison between simulated annealing and test subjects. Green circles represent the human subject's performance, black crosses indicate if the subject did not complete the task in time. Red circles show the results of the simulated annealing search algorithm, and the black line is a 3rd degree polynomial fit to the simulated annealing results.

lated annealing is based on a random search process that relies on large numbers of function evaluations. Both the simulated annealing algorithm and the test subjects' performance confirmed that taking very small steps during the search process would lead to more iterations needed to solve the problem.

This result also revealed a classic difference between the way human designers and computer based design tools are used to address design problems. Simulated annealing takes advantages of a computer's computational power to find a solution, while for the human subjects' knowledge about the system, experiences and training drives the ability of designers to find the target solution in fewer iterations.

Comparison of the subject's approaches to simulated annealing reveals that the top ranked subjects exhibited characteristics in their strategies that were strikingly similar to fine-tuned simulated annealing algorithms: having relatively large search neighborhood, decreasing temperature profile and neighborhood size. In comparison, the bottom ranked subjects tended to select small search neighborhoods that limited their mobility around the design space, and maintained a constant temperature profile that is similar to a random walk.

CONCLUSIONS AND FUTURE WORK

The work presented in this paper was motivated by the current design process used in desalination industries, which was similar to the process of solving parameter design problems. Past studies have revealed the inefficiencies of human designers when solving parameter design problems in a non-technical context. The objective of this study was to evaluate the effects of different

7

Copyright © 2015 by ASME

strategies that human designers use to solve complex engineering design problems.

The experimental study revealed a common phenomenon of subjects consistently going away from the target design, which is very similar to the behaviors of the simulate annealing optimization algorithm. Test subject's "temperature" and "neighborhood size" were estimated based on the experimental data log. It was revealed that the top ranked subjects exhibited characteristics that were consistent with a well-tuned simulated annealing algorithm, while the bottom ranked subjects are characterized by random walk around the design space. This finding is consistent with existing work that suggested that experienced designers follow a structured evaluation process [23], and use a breadth-first search method while novice designers follow a depth-first search method [24].

This study attempted to examine the process of designing desalination systems from the point of view of the human designers. There are some limitations associated with this study, for example, all test subjects were from academic settings and there was no test subject with more than 5 years of experiences working in the desalination industry, Although the results of the experimental study are specific to the reverse osmosis design problem, they suggested that human designers and computer optimization algorithms may have similarities that were previous unidentified. This finding could potentially suggest how computer based design tools should be improved in the future to amplify the strengths of humans and hide their weaknesses. For example, an interactive guidance system can be implemented in computer design interfaces to remind designers to take large step sizes and explore design spaces randomly.

The interaction and trade-off between human designers and computer design tools have not be fully understood, and warrant additional investigation. Future studies should test on a broader collection of engineering applications, as well as non-standard and ill-defined design problems. Another area of future research is to obtain a deeper, systematic understanding of the human designer's thought process through modeling and simulation, which could inspire novel design automation technologies.

The findings of the experiment could also have potential impact on the training and testing of future systems designers in universities and work places. Universities today focus on the development of component-level thinking abilities of engineering and design students, but there needs to be stronger emphasis on systems level thinking as products and systems grow in complexity. Computer design interfaces, similar to the one used in this study, are important tools to evaluate designers' system level understanding in certain engineering applications. Future implementations of this design interface could be used in universities to test their effectiveness in teaching students system-thinking approaches, and also by employers to evaluate designer's abilities in the hiring process.

REFERENCES

[1] Venkataraman, S., and Haftka, R. T., 2004. "Structural optimization complexity: what has moores law done for us?". *Structural and Multidisciplinary Optimization, 28*(6), pp. 375–387.

[2] Wang, G. G., and Shan, S., 2007. "Review of metamodeling techniques in support of engineering design optimization". *Journal of Mechanical Design, 129*(4), APR, pp. 370–380.

[3] Simpson, T. W., and Martins, J. R., 2011. "Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop". *Journal of Mechanical Design, 133*(10), p. 101002.

[4] Zhang, X., Simpson, T., Frecker, M., and Lesieutre, G., 2012. "Supporting knowledge exploration and discovery in multi-dimensional data with interactive multiscale visualisation". *Journal of Engineering Design, 23*(1), pp. 23–47.

[5] Yu, B. Y., 2014. "Human-centered approaches to system level design with applications to desalination". PhD thesis, Massachusetts Institute of Technology.

[6] Hirschi, N., and Frey, D., 2002. "Cognition and complexity: an experiment on the effect of coupling in parameter design". *Research in Engineering Design, 13*(3), pp. 123–131.

[7] Flager, F., Gerber, D. J., and Kallman, B., 2014. "Measuring the impact of scale and coupling on solution quality for building design problems". *Design Studies, 35*(2), pp. 180–199.

[8] Grogan, P., 2014. "Interoperable simulation gaming for strategic infrastructure systems design". PhD thesis, Massachusetts Institute of Technology.

[9] Stump, G., Lego, S., Yukish, M., Simpson, T. W., and Donndelinger, J. A., 2009. "Visual steering commands for trade space exploration: User-guided sampling with example". *Journal of Computing and Information Science in Engineering, 9*(4), p. 044501.

[10] Balling, R., 1999. "Design by shopping: A new paradigm?". In Proceedings of the Third World Congress of

structural and multidisciplinary optimization (WCSMO-3), Vol. 1, pp. 295–297.

[11] Austin-Breneman, J., Honda, T., and Yang, M. C., 2012. "A study of student design team behaviors in complex system design". *Journal of Mechanical Design,* **134**(12), p. 124504.

[12] Ligetti, C., Simpson, T. W., Frecker, M., Barton, R. R., and Stump, G., 2003. "Assessing the impact of graphical design interfaces on design efficiency and effectiveness". *Journal of Computing and Information Science in Engineering,* **3**(2), pp. 144–154.

[13] Simpson, T. W., Iyer, P., Rothrock, L., Frecker, M., Barton, R. R., Barron, K. A., and Meckesheimer, M., 2005. "Metamodel-driven interfaces for engineering design: impact of delay and problem size on user performance". In Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference, Austin, TX, AIAA, AIAA-2005-2060.

[14] Simpson, T. W., Barron, K., Rothrock, L., Frecker, M., Barton, R. R., and Ligetti, C., 2007. "Impact of response delay and training on user performance with text-based and graphical user interfaces for engineering design". *Research in Engineering Design,* **18**(2), pp. 49–65.

[15] Bilton, A. M., and Dubowsky, S., 2014. "A computer architecture for the automatic design of modular systems with application to photovoltaic reverse osmosis". *Journal of Mechanical Design,* **136**(10), p. 101401.

[16] Vince, F., Marechal, F., Aoustin, E., and Breant, P., 2008. "Multi-objective optimization of ro desalination plants". *Desalination,* **222**(1), pp. 96–118.

[17] Banchik, L. D., Sharqawy, M. H., and Lienhard V, J. H., 2014. "Effectiveness-mass transfer units (ε-mtu) model of a reverse osmosis membrane mass exchanger". *Journal of Membrane Science,* **458**, pp. 189–198.

[18] Koul, H., Susarla, V., and Van Ryzin, J., 1981. "Regression analysis with randomly right-censored data". *The Annals of Statistics*, pp. 1276–1288.

[19] Peto, R., and Peto, J., 1972. "Asymptotically efficient rank invariant test procedures". *Journal of the Royal Statistical Society. Series A (General)*, pp. 185–207.

[20] Newton, E., and Rudel, R., 2007. "Estimating correlation with multiply censored data arising from the adjustment of singly censored data". *Environmental science & technology,* **41**(1), pp. 221–228.

[21] Dowsland, K. A., and Thompson, J. M., 2012. "Simulated annealing". In *Handbook of Natural Computing*. Springer, pp. 1623–1655.

[22] Goldstein, L., and Waterman, M., 1988. "Neighborhood size in the simulated annealing algorithm". *American Journal of Mathematical and Management Sciences,* **8**(3-4), pp. 409–423.

[23] Ahmed, S., Wallace, K. M., and Blessing, L. T., 2003. "Understanding the differences between how novice and experienced designers approach design tasks". *Research in Engineering Design,* **14**(1), pp. 1–11.

[24] Ball, L. J., St. BT Evans, J., Dennis, I., and Ormerod, T. C., 1997. "Problem-solving strategies and expertise in engineering design". *Thinking & Reasoning,* **3**(4), pp. 247–270.

[25] Goffe, W. L., Ferrier, G. D., and Rogers, J., 1994. "Global optimization of statistical functions with simulated annealing". *Journal of Econometrics,* **60**(1), pp. 65–99.

Appendix A: Simulated Annealing

Simulated annealing is a heuristic algorithm that searches for an optimal design (minimizing an objective function *J*) by iteratively exploring the design space and moving both uphill and downhill. The algorithm is inspired by the annealing process for physical materials like metal and polymers, where slow cooling (annealing) allows the material to arrive at the lowest energy state. Inherent random fluctuations in energy allows the annealing system to escape local energy minima to achieve the global minimum.

At each iteration, the algorithm randomly picks a design vector $\mathbf{x}'$ in a neighborhood around the current design vector $\mathbf{x}$:

$$x_i' = x_i + v_i \cdot r \tag{1}$$

where x_i', x_i are elements of design vectors $\mathbf{x}'$ and $\mathbf{x}$. v_i indicates the neighborhood size, and r is a random numbers from [-1, 1]. The objective function value $J(\mathbf{x}')$ is evaluated and compared to $J(\mathbf{x})$.

If $J(\mathbf{x}')$ is less than $J(\mathbf{x})$, then $\mathbf{x}'$ is accepted, $\mathbf{x}$ is set to $\mathbf{x}'$ for the next iteration, and the algorithm moves downhill. If $J(\mathbf{x}')$ is greater than $J(\mathbf{x})$, then acceptance is determined based on the probability described in Equation 2 [25]:

$$P_i = \frac{1}{1 + e^{\Delta_i / T_i}} \tag{2}$$

where P_i is the probability of accepting a worse design at iteration i, Δ_i is the differences between the current iteration's objective function value and previous iteration's, and T_i is the temperature of the algorithm at iteration i. A high temperature value corresponds to a higher probability of accepting a worse design (moving uphill), while a low temperature means a low probability of accepting a worse design. Typically the temperature of a simulated annealing algorithm decreases following an exponential cooling schedule, and the search neighborhood is

also reduced over time proportional to the temperature. A decreasing temperature profile (cooling schedule) ensures that the design space is explored randomly at first, and then focuses on improving the design toward the end of the search.