

# Reducing Human Fatigue in Interactive Evolutionary Computation through Fuzzy Systems and Machine Learning Systems

Kamalian, Raffi R.  
Kyushu University

Yeh, Reic

Zhang, Ying

Agogino, Alice M.

他

<https://hdl.handle.net/2324/1807777>

---

出版情報 : IEEE International Conference on Fuzzy Systems. 2006, pp.678-684, 2006-07-16. IEEE  
バージョン :  
権利関係 :



# Reducing Human Fatigue in Interactive Evolutionary Computation through Fuzzy Systems and Machine Learning Systems

Raffi Kamalian, Eric Yeh, Ying Zhang, Alice M. Agogino, and Hideyuki Takagi, *Member, IEEE*

**Abstract**— We describe two approaches to reducing human fatigue in Interactive Evolutionary Computation (IEC). A predictor function is used to estimate the human user's score, thus reducing the amount of effort required by the human user during the evolution process. The fuzzy system and four machine learning classifier algorithms are presented. Their performance in a real-world application, the IEC-based design of a micromachine resonating mass, is evaluated. The fuzzy system was composed of four simple rules, but was able to accurately predict the user's score 77% of the time on average. This is equivalent to a 51% reduction of human effort compared to using IEC without the predictor. The four machine learning approaches tested were k-nearest neighbors, decision tree, AdaBoosted decision tree, and support vector machines. These approaches achieved good accuracy on validation tests, but because of the great diversity in user scoring behavior, were unable to achieve equivalent results on the user test data.

## I. INTRODUCTION

We present an improvement on conventional Interactive Evolutionary Computation (IEC) as applied to the field of machine design. In previous work we have developed and tested an IEC-based approach to the design of micromachines, also known as microelectromechanical systems (MEMS). This paper describes our work on the use of fuzzy inference systems and machine learning techniques to reduce human fatigue, a limiting factor in the field of IEC.

### A. Evolutionary Design Synthesis

Evolutionary Computation (EC) is an optimization and search method based on the principles of natural or biological evolution and optimizes target tasks using evolutionary operations such as selection, crossover, and mutation. Unlike many traditional optimization techniques, EC is a population-based search method and does not require gradient information to search. It only requires the ability to

encode and alter a proposed design and the ability to evaluate its performance. EC is also well suited for design synthesis – finding a valid design that meets the desired performance goals as closely as possible without violating any constraints. The Genetic Algorithm (GA) is the most popular form of EC and is the approach used in this research [1].

One key limitation of a conventional EC approach is that objectives and constraints must be clearly defined numerically and explicitly included in the EC formulation. In the real world of machine design, there are often many factors that experienced designers take into account but cannot be explicitly expressed as a numerical fitness equation for EC. One solution to this problem is to involve the designer actively during the evolution process.

### B. Human Interactive Evolution

Interactive Evolutionary Computation is a method for optimizing a system using subjective human evaluation as part of an evolutionary search [2]. It is well suited for optimizing systems whose evaluation criteria are preferential or subjective, such as graphics, music and design, and systems that can be evaluated based on an expert's domain knowledge. Primarily, this approach is applied to aesthetic or sensory-perception-related design. Applications include computer graphics, music, industrial design, architecture, sound and image processing, virtual reality, data mining, and other engineering and educational applications.

A key limitation of IEC is human fatigue. While computers evaluating numerical objectives do not get tired, human beings reviewing and rating designs via a graphical user interface (GUI) do experience fatigue. This limits the population size or number of generations over which evolution can take place, effectively limiting the amount of searching of the design space that can be performed.

For engineering design, IEC can be used to augment the numerical objectives and constraints, which allows the human user to include design experience and judgment to steer the evolution process. Another benefit of IEC is the ability of the human to adapt their subjective evaluation through the course of the evolution and discover features of interest that had not been anticipated in advance when the traditional EC constraints and objectives were coded. For this reason, interactive evolution has been referred to as a 'novelty generator' [3].

### C. Design of MEMS Using Evolutionary Approaches

Multiobjective GA-based design synthesis for MEMS was

This work was supported in part through US NSF grant CCR-DES/CC-0306557, the JSPS Postdoctoral Fellowship program and Grant-in-Aid for Scientific Research.

Raffi Kamalian is a JSPS post-doctoral fellow at Kyushu University, Fukuoka 815-8540, JAPAN (phone: +81-92-553-4555; e-mail: raffi@design.kyushu-u.ac.jp).

Eric Yeh is with SRI International, Palo Alto, CA 94025 USA. (e-mail: yeh@ai.sri.com).

Ying Zhang is a PhD candidate in Systems Engineering at the University of California, Berkeley, CA 94720, USA. (e-mail: yzh@berkeley.edu).

Alice M. Agogino is on the faculty of the Mechanical Engineering Department at the University of California, Berkeley, CA 94720, USA (e-mail: agogino@berkeley.edu).

Hideyuki Takagi is with the Faculty of Design, at Kyushu University, Fukuoka 815-8540, JAPAN (e-mail: takagi@design.kyushu-u.ac.jp).

first introduced by Zhou *et al.* [4]. Multiobjective Genetic Algorithms (MOGAs) are well suited to the multiple design goals and constraints and limitations inherent in MEMS. This initial work has since been extended and applied to several real-world MEMS design applications including MEMS accelerometers and gyroscopes. An example of a MEMS resonating mass synthesized with GA can be seen in Fig. 1.

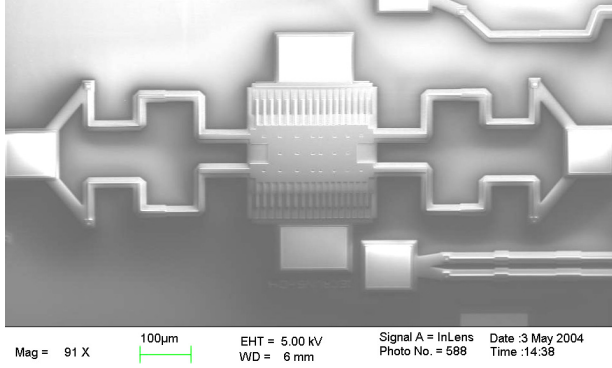


Fig. 1. MEMS resonating mass synthesized using a Multiobjective Genetic Algorithm

The first application of IEC to MEMS was a traditional IEC implementation applied to refine the output of a conventional MOGA [5]. Building on the observations of our user test of this approach, we developed a novel supervisory IEC tool, where the human reviews designs every  $n^{th}$  generation of a conventional MOGA [6]. This automated evolution with occasional human ‘review’ combines the tirelessness and speed of the computer with the more ‘expensive’, in terms of time and fatigue, opinion of the human (Fig. 2).

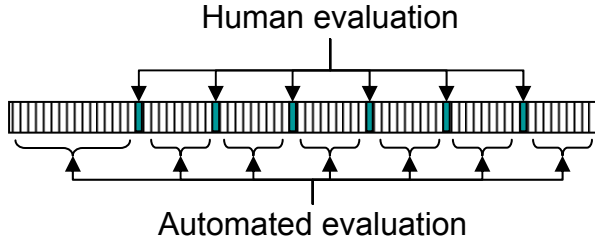


Fig. 2. Schematic of interspersed human interaction during automated evolutionary synthesis, used for MEMS design synthesis.

Zhou’s original MOGA application to MEMS design utilizes Pareto ranking to derive the fitness of a design in a multiobjective design space. This fitness is then used to establish the probability that a given design will pass its genetic material on to the next generation. In order to mesh with this method of evaluating fitness with human interaction, in [5] we proposed the concept allowing the human to perform Pareto rank shifting.

Rather than giving a numerical score (e.g., 1-5) as in our first IEC approach, in this supervisory IEC, the human user

has the option to shift the rank of a given design up or down. In the GUI (Fig. 3), the user is presented with three radio buttons for each design: ‘promote’, ‘~’ (neutral) and ‘demote’. Selecting ‘demote’ for a design shifts its rank downward, selecting ‘promote’ shifts the design’s rank upward, and ‘~’, referred to as neutral score, does not shift the rank of the design.

A benefit of this scoring approach is that it no longer requires the human to score each and every design, reducing human fatigue significantly. Any additional reductions would help improve the tool further.

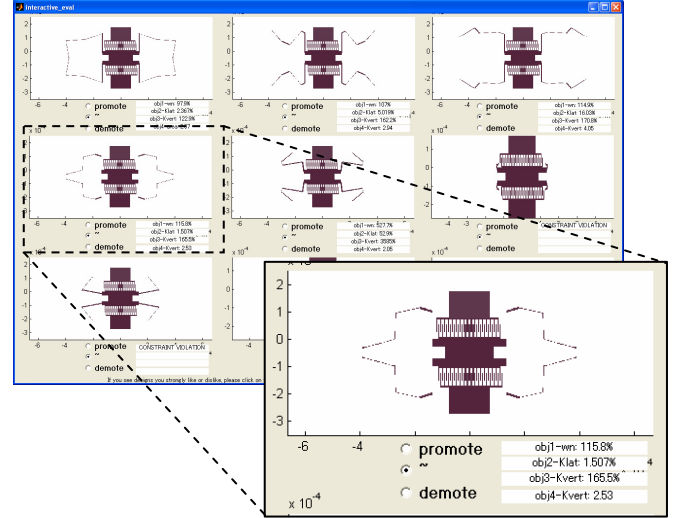


Fig. 3. User interface of the supervisory IEC tool. The user can guide the evolution process by giving a rank shift upward (promote) or downward (demote) if desired. The user also has the option of not shifting the rank (denoted by the ‘~’ in the GUI). Our goal is to preselect the most likely rank shift choice for each design when it is presented to the user.

#### D. Human Evaluation Prediction

One method of further reducing the effort required by a human is the use of artificial intelligence techniques to predict the human user’s scoring of a design based on previous scoring. While it is hard to quantify all the factors that humans take into account when making their evaluation, a few factors can be extracted from previous user tests. Several factors, based on the geometry of the designs, were present in many of the designs the human chose to demote, such as very acute beam intersections, high lateral stiffness (legs with little meandering), very short beams, or beams that came very close to intersecting. Kamalian *et al.* [7] give more information as to why these features are undesirable to MEMS designers based on results from fabrication and testing. Fig. 4 shows two examples.

Fuzzy systems and machine learning systems are methods of generating a transfer function between a feature space and a given class [8]. In MEMS applications, the feature space can be considered the geometric or performance features of a MEMS design, and the classification is the score given to that design. These methods can be used to create a prediction

function for IEC. Predicting and preselecting the human score on the IEC user interface reduces the effort required by the human. The human's responsibility shifts from scoring each design to simply validating that the predictor's score matches the human's own and correcting it when it does not match.

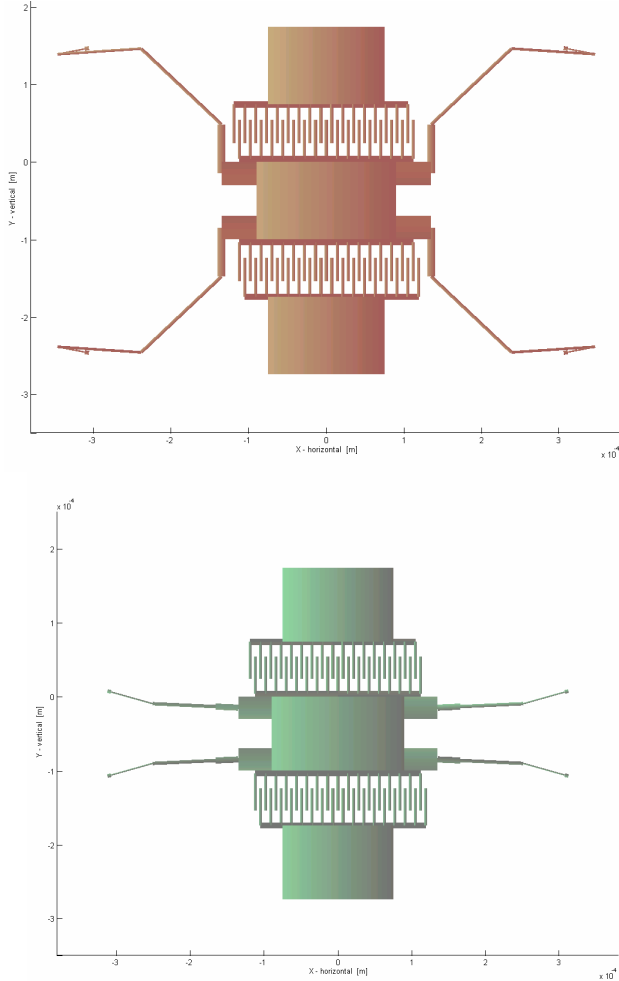


Fig. 4. Two examples of designs commonly given a ‘demote’ ranking by the human user. The top design contains a very acute beam intersection at the end of the legs, which is undesirable to MEMS designers; the bottom design has very high lateral stiffness, which will cause performance problems after fabrication

This concept of learning the human's scoring patterns and trying to predict them has also been used to create an automated ‘agent’ that scores in place of a human user. This agent, residing in software, has the ability to evaluate many more designs in a much shorter time than the human from whom it learned. This approach has been presented [9],[10] using a neural network or RBF network to learn human preference. Gu *et al.* [3] present a review of several fitness approximation approaches in their work in this area, which they term ‘capturing implicit aesthetic intention.’

In this paper we present the design and application of a predictor created using a fuzzy inference system. This predictor is used on the MEMS resonator design example

presented in [6]. User tests are presented to show the effectiveness of this approach. We also look at applying four machine learning approaches – k-nearest neighbors, decision trees, AdaBoost, and support vector machine – and compare their effectiveness. Finally, the results are summarized and recommendations for future research are made.

## II. FUZZY SYSTEM FOR IEC PREDICTION

Since its introduction by Zadeh in 1965 [11], fuzzy logic has become an important tool for control, diagnosis, knowledge representation, and other expert systems. Fuzzy logic allows intermediate membership values to be defined other than the conventional Boolean evaluations of true/false, yes/no, or on/off. Fuzzy inference systems map from one or more inputs to one or more outputs, connecting human-based semantics with computer programming.

### A. Fuzzy Predictor Description

In this application, we are looking to automatically shift the score on designs that can be easily predicted. Through our user studies in [5] and [6], we observed reoccurring features that prompted the users to quickly demote a design – very acute beam intersections, very narrow leg spacing, very high stiffness ratio, and very ‘stubby’ designs. While no one feature would necessarily condemn a design, the human user generally made a judgment call based on the severity and combination of these features.

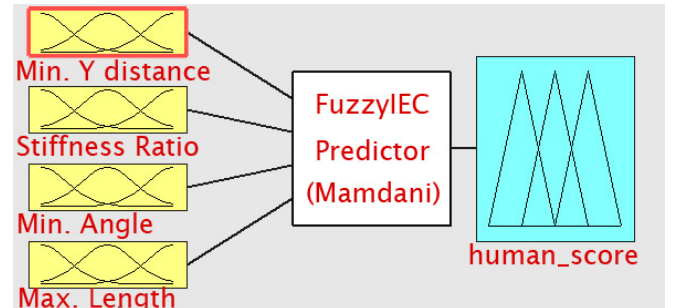


Fig. 5. Schematic of fuzzy system, implemented with the MATLAB fuzzy toolbox

TABLE I  
RULES OF FUZZY SYSTEM

rule	if	then
1	<i>stiffness-ratio is too-stiff</i>	<i>demote</i>
2	<i>min-angle is too-acute</i>	<i>demote</i>
3	<i>min-y is too-close</i>	<i>demote</i>
4	<i>max-length is too-stubby</i>	<i>demote</i>

These rules can be compiled into the four simple fuzzy rules presented in Table I. The rules were coded into a Mamdani-type fuzzy inference system using the MATLAB fuzzy toolbox. Fig. 5 shows the schematic representation, and Fig. 6 shows the membership functions corresponding to each of the input variables as well as the output function.

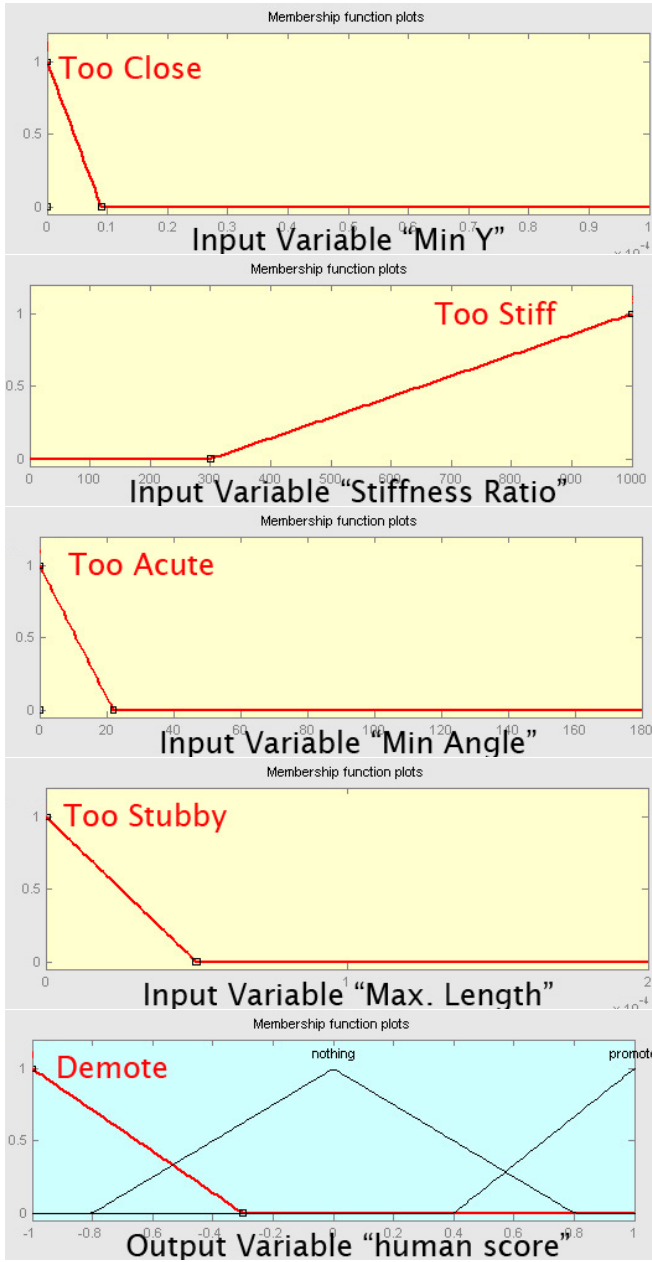


Fig. 6. Membership functions of fuzzy system. Each membership function corresponds to a rule in Table I. The membership for the output function is also presented.

While very simple single variable rules could be generated for the ‘demote’ situation, it was much harder to get a good definition of what prompts a human to ‘promote’ a design. It is harder to classify the features human designers find desirable, and prompt them to want to make this shape more likely to reproduce the evolution. This promotion was much rarer in our user tests, however, only occurring approximately 8% of the time a human rates a design on average (meaning only about 1 or 2 promote scores per generation). Based on this small percentage, even if a predictor cannot identify a ‘promote’ design, it can be sufficiently beneficial to our goal.

In the user test data, on average 37% of the designs are scored ‘demote’, and the remaining 56% are neutral – neither ‘demote’ nor ‘promote’. Based on these ratios, for our first

attempt at a fuzzy predictor we focused on choosing between ‘demote’ (negative rank shift) or ‘neutral’ (no rank shift).

The membership functions presented in Fig. 6 were tuned on a set of 927 training designs generated in the 20<sup>th</sup> generation of eight separate IEC runs and rated by a single human user (one of the authors). On this aggregate set of designs and corresponding human scores, the tuned fuzzy system matched the human score on 674 designs (72.7% accuracy). In this data set, 552 designs (60%) were scored ‘neutral’ by the human, meaning 375 were scored either ‘promote’ or ‘demote’. By comparing the accuracy of the fuzzy predictor over simply giving a default of ‘neutral’ to all designs (as was done in [6]), we can gauge the effectiveness in reducing human fatigue. *Human effort* required, defined as mouse clicks to change the radio buttons in the GUI to the human’s desired values, is reduced from 375 rescores required if ‘neutral’ is selected for all designs to 253 rescores when the fuzzy predictor is used. This is equivalent to a 32% reduction in *human effort* if this fuzzy system were to be used as a predictor for this data set.

### B. Fuzzy Predictor User Validation Test

An important question that remains is the applicability of one human’s preference to another human’s. In aesthetic and sensory cases, the concept of ‘good’ versus ‘bad’ can vary greatly (“one man’s trash is another man’s treasure”), but in cases of engineering design, some factors are more fundamental: these can be expressed as ‘rules of thumb’ or ‘good design practice.’ The relative weighting and cut-off points of these rules may vary from engineer to engineer, but we would like to determine if the fuzzy system can be used to build a common predictor, useful to all users.

TABLE II  
THE ACCURACY OF THE FUZZY PREDICTOR, ALONG WITH THE HUMAN EFFORT THAT WAS SAVED FOR THE NINE USER TESTS CONDUCTED. USERS WITH MEMS EXPERIENCE BENEFITED MOST FROM THE USE OF THE PREDICTOR

User Test	MEMS expert?	Number of designs rated	Accuracy of prediction
1	Y	154	80%
2	Y	153	85%
3	Y	147	87%
4	Y	162	85%
5	Y	130	89%
6	Y	131	86%
7	N	117	50%
8	N	135	70%
9	N	152	61%
Average			77%
Standard deviation			6.5%

We conducted nine test runs (from a total of six users). The same optimization problem used in [6] was repeated, with the same GA settings. The only difference was the use of the fuzzy predictor to preselect the radio buttons next to each

design in the IEC GUI. Table II presents the results. The data shows that on average the predictor achieved a 77% accuracy level, reducing the *human effort* by 51% on average (Table III). The scoring habits varied widely between users, some were very active in adjusting rank (a very high number of promote/demote rankings made) while others were more passive and chose not to adjust the Pareto ranking often.

TABLE III  
HUMAN EFFORT SAVED, DEFINED AS THE AMOUNT OF HUMAN EFFORT  
(RATING CHANGES IN THE IEC GUI) REDUCED BY USE OF THE FUZZY SYSTEM

User Test	MEMS expert?	# of incorrect predictions made	# of user promote /demote ratings made	Effort Saved
		(effort with predictor)	(effort without predictor)	
1	Y	30	66	55%
2	Y	23	52	56%
3	Y	19	53	64%
4	Y	25	53	53%
5	Y	14	48	79%
6	Y	18	45	60%
7	N	58	72	19%
8	N	40	71	44%
9	N	59	102	42%
<b>Avg</b>		<b>32</b>	<b>62</b>	<b>51%</b>
Std dev		17	18	15%

Although there are not enough users to draw statistically significant conclusions, it is worth noting the difference in performance for users with MEMS design experience and those without. Experienced users achieved much higher matching, 85% on average, with a 60% reduction in effort, whereas the accuracy and effort saved averages for nonexperts were 60% and 35%, respectively. Despite the differences between the user types, the fuzzy predictor reduced the *human effort* required in all nine cases.

This result is very positive; a 51% reduction in *human effort* means useful designs can be synthesized with less impact on the user, or larger populations or more generations of evolution can be used to better search the design space. It also shows that there is some fundamental correspondence to the design judgment of engineers.

It is important to note that the predictor achieved better matching in many of the user tests than it did for the original training set on which it was tuned. It is possible that this is a case of “the cart leading the horse.” Some users may tend to trust the predictor’s decision and are hesitant to change it unless they strongly disagree. This has advantages and disadvantages. The predictor may be reducing the human’s influence by providing a prediction in advance, or it might aid the human by giving a baseline of engineering rules of thumb from which to work.

### C. Mapping Promote Rating from User Test Results

After collecting the user results, a further attempt was made to extract fuzzy rules to describe the ‘promote’ rating. Unfortunately for the four inputs used by the current system there is an easily derived membership function or set of functions that maps only to designs rated ‘promote’. For example one unifying factor in all the designs scored ‘promote’ in user test #2 was a very low stiffness ratio. This factor is present in many designs rated ‘demote’ or ‘neutral’ as well. This reduction in prediction accuracy also existed for various combinations of variables (e.g., *if low stiffness AND obtuse angles then promote*). It is possible that the factors prompting the human to score ‘promote’ are not related directly to the factors that lead the human to score ‘demote’. If we want to add the ability to predict ‘promote’ cases, a change in fuzzy system input may be necessary.

## III. BASIC MACHINE LEARNING FOR IEC PREDICTION

Using the same set of 927 training designs as described in the previous section, four machine learning classification methods were tested: k-nearest neighbors, decision tree, AdaBoosted decision tree, and Support Vector Machine (SVM). In addition, decision stumps, with and without AdaBoost, were tested. The input vector was the geometry for a design – the length, width, and angle information for all beams in the design (as many as seven beam segments per leg). These values were normalized to the 0–1 range, and each input vector was labeled with the human score of *promote*, *demote*, or *neutral* (indicating no opinion). Each method was validated using a set of 347 designs generated and scored in a manner identical to the training set. The accuracy of each method was then tested on the designs and scores generated in the user tests presented in the previous section.

### A. Learning Systems Introduction

The k-nearest neighbors method classifies a test vector as the majority labeling of the *k* most similar vectors to that vector from the training set [12]. A decision tree is a branching tree structure that obtains classifications via a feature-based traversal of its nodes. For this experiment, the C4.5 algorithm was used to induce the decision trees [13].

AdaBoost combines multiple learners in sequence to form a unified classifier that exhibits stronger performance than its component learners [14]. For this experiment, these learners were C4.5 decision trees. To offer a contrast, decision stumps, decision trees with depth one were tested both singly and in AdaBoosted form.

SVM is a method that creates classifications by determining hyperplanes that maximize the margins between different classes in the training data [15],[16]. For the experiment, an SVM with a radial basis kernel was implemented using the LibSVM Library [17].

TABLE IV  
PERFORMANCE OF SIX MACHINE LEARNING CLASSIFIERS ON A VALIDATION DATA SET. THE PERCENTAGE OF NEUTRAL SCORES PRESENT IN THE DATA SET IS AT THE BOTTOM. IMPROVEMENT OVER THIS IS USED TO CALCULATE THE ‘EFFORT SAVED’ IF THIS METHOD IS USED AS A PREDICTOR

Method	Validation Accuracy	Effort Saved
k-Nearest Neighbors	72.9%	18.4%
Decision Tree	70.7%	11.8%
AdaBoost, Decision Tree	74.2%	22.4%
Decision Stump	66.8%	0%
AdaBoost, Decision Stumps	68.1%	3.95%
SVM	74.7%	23.7%
Neutral score (baseline)	66.8%	

TABLE V  
CLASSIFIER PERFORMANCE OF SIX MACHINE LEARNING CLASSIFIERS OVER DATA GENERATED FROM USER TESTS. OF THE SIX CLASSIFIERS TESTED, ADABOOST DECISION TREE PERFORMED BEST.

Method	Avg. User Test Accuracy	Accuracy Std-Dev	Avg. Effort Saved	Effort Saved Std-Dev
k-Nearest Neighbors	63.6%	11.7%	18.8%	6.9%
Decision Tree	63.8%	12.4%	19.5%	11.9%
AdaBoost, Decision Tree	67.8%	14.1%	30.2%	11.6%
Decision Stump	55.5%	12.1%	0%	0%
AdaBoost, Decision Stumps	65.7%	11.3%	23.9%	7.1%
SVM	65.5%	12.7%	23.9%	9.0%

Optimal models and parameters for each of the classification schemes were determined using the validation set, and these were tested against the geometry and user scores generated in user tests 1 and 2. Decision trees were grown to depth 5, k-nearest neighbors used a  $k$  value of 83, and the SVM used the default LibSVM settings with a penalty parameter of 4. The AdaBoosted algorithms each used 60 base learners, with the optimal boosted decision tree depth found at 3.

### B. Learning System Results

Overall, the AdaBoosted decision tree method ranked relatively high in the validation set (Table IV) and proved to have the best generalization performance across the user tests (Table V). Most of the methods generally displayed a saving in user effort, with the exception of the non-AdaBoosted decision stump, which essentially labeled every example as *neutral*.

Given their geometric nature, decision trees can be expressed as a series of rule/predicate applications [12].

AdaBoosted decision trees can be viewed as being a method of combining these rules in a manner that can generalize well [14]. Given that, and the strong performance of the fuzzy rules set, it is likely that approaches that discover and employ combinations of rules and predictions can lead to good results in future studies.

TABLE VI  
DISTRIBUTION OF HUMAN CLASSIFICATIONS (PROMOTE/DEMOTE/NO SCORE) ACROSS VALIDATION SET AND USER TEST SETS.

Data Set		Promote	Demote	Neutral
Validation		3.5%	29.7%	66.8%
User test #	1	5.2%	37.7%	57.1%
	2	9.8%	24.2%	66.0%
	3	4.8%	31.3%	63.9%
	4	2.5%	30.2%	67.3%
	5	0.0%	36.9%	63.1%
	6	4.6%	32.1%	63.4%
	7	14.5%	47.0%	38.5%
	8	7.4%	45.2%	47.4%
	9	21.1%	46.1%	32.9%
Average		7.8%	36.7%	55.8%

## IV. FUTURE WORK

### A. Fuzzy Predictor Improvement

Because of its ability to connect human-based semantics with computer programming, the fuzzy system has a distinct benefit over other AI techniques. The addition of any rules that improve the performance of a fuzzy system can help identify useful expert domain knowledge. As mentioned in section II, deriving more accurate fuzzy rules manually is difficult, but automatic methods exist that may help with this.

A neuro-fuzzy system is a combination of fuzzy systems and neural networks, and one of such combinations is a fuzzy system that is auto-tuned by neural network learning [18]. The neural network’s learning capability can automatically generate additional rules and membership functions, thus providing means of identifying what rules are being used by the human users. If additional inputs that influence the user’s decisions can be identified, a neuro-fuzzy system can help in understanding their impact and incorporate it into the predictor.

### B. Machine Learning Predictor Improvement

Given the results presented in section III, we would like to investigate methods to induce combinations of features and rules that better classify input vectors, whether it is through statistical model formation or a combination bootstrapped with a base of knowledge.

Since the users exhibited significant differences in labeling across the group (Table VI), specializing on individual users and generalizing across them is another avenue to explore. In addition, online learning mechanisms can be leveraged to exploit the sequential nature of the data.



## V. CONCLUSION

We have investigated two methods for creating a predictor function to reduce the human fatigue associated with interactive evolution. The fuzzy-system-based predictor was based on four system rules derived manually from the observation of previous human user tests. Despite a wide variation in user behavior, the fuzzy predictor matched the human's score for 77% of the designs rated, reducing the amount of effort required by the human during the evolutionary process for all nine user tests performed. The average reduction in human effort was 51%. This result validates the use of fuzzy systems in an IEC MOGA design tool.

Four machine learning algorithm approaches were also investigated. While all achieved some reduction of effort, with 68% accuracy and a reduction in human effort of 31%, none achieved the same level of performance as the fuzzy predictor.

## REFERENCES

- [1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman, Boston, MA, 1989.
- [2] H. Takagi, "Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation", *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275-1296, 2001.
- [3] Z. Gu, M. Tang, and J.H. Frazer, "Capturing Aesthetic Intention during Interactive Evolution", *Computer-Aided Design*, 1-14, 2005.
- [4] N. Zhou, B. Zhu, A. M. Agogino, and K.S.J. Pister, "Evolutionary Synthesis of MEMS (Micro Electronic Mechanical Systems) Design", *Proceedings of the Artificial Neural Networks in Engineering (ANNIE2001)*, pp. 197-202, 2001.
- [5] R. Kamalian, H. Takagi, and A.M. Agogino, "Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation", *Proceedings of GECCO 2004, Genetic and Evolutionary Computation Conference*, pp. 1030-1041, 2004.
- [6] R. Kamalian, Y. Zhang, H. Takagi, and A.M. Agogino, "Reduced Human Fatigue in Interactive Evolutionary Computation for Micromachine Design", *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, China, August 18-21 2005, pp 5666-5671.
- [7] R. Kamalian, Y. Zhang, and A.M. Agogino, "Microfabrication and Characterization of Evolutionary MEMS Resonators", *Proceedings of the 2005 IEEE International Symposium on Micro-NanoMechatronics and Human Science*, Nagoya, Japan, 2005, pp. 109-114.
- [8] B. Knapp, "Fuzzy Sets and Pattern Recognition". Available: <http://hci.sapp.org/lectures/knapp/fuzzy/fuzzy.pdf>.
- [9] I. Ecemis, E. Bonabeau, and T. Ashburn, "Interactive Estimation of Agent-Based Financial Markets Models: Modularity and Learning", *Proceedings of GECCO 2005, Genetic and Evolutionary Computation Conference*, 2005, pp.1897-1904.
- [10] D. S. Todd and P. Sen, "Directed Multiple Objective Search of Design Spaces Using Genetic Algorithms and Neural Networks", *Proceedings of GECCO 1999, Genetic and Evolutionary Computation Conference*, 1999, pp. 1738-1743.
- [11] L. Zadeh, "Fuzzy Sets", *Information and Control*, Vol. 8, 1965.
- [12] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 2001.
- [13] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993.
- [14] Y. Freund and R. Schapire, "Experiments with a New Boosting Algorithm", *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148-156, 1996.
- [15] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, NY, 1995.
- [16] C. Cortes and V. Vapnik, "Support-Vector Networks". *Machine Learning* 20, pp. 273-297, 1995.
- [17] C. Chang and C. Lin, "LIBSVM: A Library for Support Vector Machines", Available at <http://csie.ntu.edu.tw/~cjlin/libsvm>.
- [18] J.-S. R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May 1993.