PAPER

Interactive Evolutionary Computation Using a Tabu Search Algorithm

Hiroshi TAKENOUCHI[†], *Student Member*, Masataka TOKUMARU[††a)], *and* Noriaki MURANAKA[††], *Members*

SUMMARY We present an Interactive Tabu Search (ITS) algorithm to reduce the evaluation load of Interactive Evolutionary Computation (IEC) users. Most previous IEC studies used an evaluation interface that required users to provide evaluation values for all candidate solutions. However, user's burden with such an evaluation interface is large. Therefore, we propose ITS where users choose the favorite candidate solution from the presented candidate solutions. Tabu Search (TS) is recognized as an optimization technique. ITS evaluation is simpler than Interactive Genetic Algorithm (IGA) evaluation, in which users provide evaluation values for all candidate solutions. Therefore, ITS is effective for reducing user evaluation load. We evaluated the performance of our proposed ITS and a Normal IGA (NIGA), which is a conventional 10-stage evaluation, using a numerical simulation with an evaluation agent that imitates human preferences (Kansei). In addition, we implemented an ITS evaluation for a running-shoes-design system and examined its effectiveness through an experiment with real users. The simulation results showed that the evolution performance of ITS is better than that of NIGA. In addition, we conducted an evaluation experiment with 21 subjects in their 20 s to assess the effectiveness of these methods. The results showed that the satisfaction levels for the candidates generated by ITS and NIGA were approximately equal. Moreover, it was easier for test subjects to evaluate candidate solutions with ITS than with NIGA.

key words: *interactive evolutionary computation, user interface, interactive tabu search*

1. Introduction

Interactive Evolutionary Computation (IEC) is a method used to support the development of a product. IEC uses a computer to employ affective engineering (also known as Kansei engineering) techniques to study the interactions between a product and its users. Users desire products to be functional. IEC systems can help designers in developing a product that also satisfies users' subjective and emotional preferences [1]. The Interactive Genetic Algorithm (IGA) is known as an IEC method. Researchers have studied various IEC systems, including image-retrieval systems [2], clothing-design systems [3], interior-layout systems [4], sign-sound-generation systems [5], hearing-aid-fitting systems [6], and robot-action-generation systems [7]. These studies used various methods to help users evaluate product solutions.

General methods used to perform IEC evaluation require evaluation values for each candidate solution associated with the presented product; therefore, users must compare a large number of product solutions, and when users have to compare similar candidates solution, the evaluation of each candidate becomes difficult.

In a simulated breeding IEC evaluation method, the user chooses only preferred candidate solutions from the presented candidate solutions [8]. This method involves a 2-stage evaluation. The candidates chosen by a user are assigned the value of 1, and those not chosen by a user are assigned the value of 0. The user intuitively evaluates the candidate solutions. However, this method often produces localized solutions, which make the evolution of candidate solutions difficult. Because of the limited coarse 0 or 1 evaluation, the crossover solutions are primarily those with the value of 1.

Several methods have been proposed to reduce user evaluation load in an IEC system. In particular, researchers have determined that the evaluation process is more effective when it is simplified [1]. Paired comparison and tournament-style evaluation methods, which allow users to simultaneously choose between two candidate solutions, are more effective than methods that employ a 5- or 10-stage evaluation [9]–[12]. However, these methods require repeated evaluation of the same candidate solution.

In this study, we demonstrate the effectiveness of an IEC evaluation interface where users select their favorite candidate solution from the presented candidate solutions. This interface can simplify the evaluation process and reduce user evaluation load. However, if an IGA interface uses this evaluation interface, accurate evaluation values for candidate solutions are not generated and solution evolution is insufficient. Therefore, we propose an Interactive Tabu Search (ITS) that uses Tabu Search (TS) as an Evolutionary Computation (EC) algorithm. The effectiveness of TS as an optimization method has already been reported [13].

Researchers have proposed method to apply TS to IEC candidate solutions retrieval [14], [15]. Klau et al. have proposed a human-guided TS method covering the Traveling Salesman Problem (TSP) and the Scheduling Problem (SP) as interactive optimization problems [14]. Munemoto et al. have proposed an interactive multi-objective optimization method using TS for a floor layout [15]. However, these methods do not require the user to evaluate the candidate solutions for TS. Our proposed ITS method applies a new evaluation interface to IEC solution evaluation in which the user chooses their favorite candidate solution from the pre-

Manuscript received May 11, 2012.
Manuscript revised October 30, 2012.
[†]The author is with the Graduate School of Science and Engineering, Kansai University, Suita-shi, 564–8680, Japan.
[††]The authors are with the Faculty of Engineering Science, Kansai University, Suita-shi, 564–8680 Japan.
a) E-mail: toku@kansai-u.ac.jp
DOI: 10.1587/transinf.E96.D.673

sented candidate solutions.

ITS then generates neighboring candidate solutions from the current optimum solution. ITS evaluation only chooses the best candidate solution from current candidate solutions. However, because ITS is a local search method, candidate solutions converge locally. Moreover, users must choose one candidate solution from the presented candidate solutions even if a preferred or favorite candidate solution does not exist. However, ITS has a simpler evaluation interface than systems using the Normal Interactive Genetic Algorithm (NIGA), which is a conventional 10-stage evaluation. ITS also possesses a greater likelihood of reducing user evaluation load.

The following two questions should be addressed to confirm the effectiveness of ITS.

1. Is the optimization performance of ITS equal to that of NIGA?
2. Does ITS have lower user evaluation load compared with NIGA?

The first question is essential when attempting to reduce user evaluation load. ITS employs a local search to find the optimum solution. If ITS finds a localized solution, it must repeat solution evaluations before arriving at the optimum solution. Therefore, we examined the effectiveness of ITS and NIGA using a numerical simulation with an evaluation agent that imitated human preferences (Kansei). In addition, we quantitatively compared the evolution performances of ITS and NIGA.

To verify the numerical simulation results, user evaluation loads should be examined through an evaluation experiment with real users. To answer the second question, we inspected the effectiveness of ITS and NIGA. The experiment evaluated the user satisfaction level for a chosen candidate and the usability of each interface.

The next section describes the TS algorithm and ITS used in our proposed method. Section 3 describes the numerical simulation for the performance evaluation of ITS and NIGA. Sections 4 and 5 describe the evaluation experiments using real users and the experimental results, respectively. Finally, Sect. 6 concludes the paper and discusses topics of future research.

2. Proposed Method

2.1 Tabu Search

TS is a local search method proposed by Fred Grover in 1989 [13]. Figure 1 illustrates TS solution retrieval. First, TS randomly generates initial candidate solutions and determines an optimum candidate solution from these initial candidate solutions. Next, TS searches a local neighborhood of the current optimum candidate solution p_1 and generates new candidate solutions. Then, TS determines the next optimum candidate solution p_2 from the new candidate solutions. TS moves to another candidate solution although the current candidate solution is a local optimum solution.

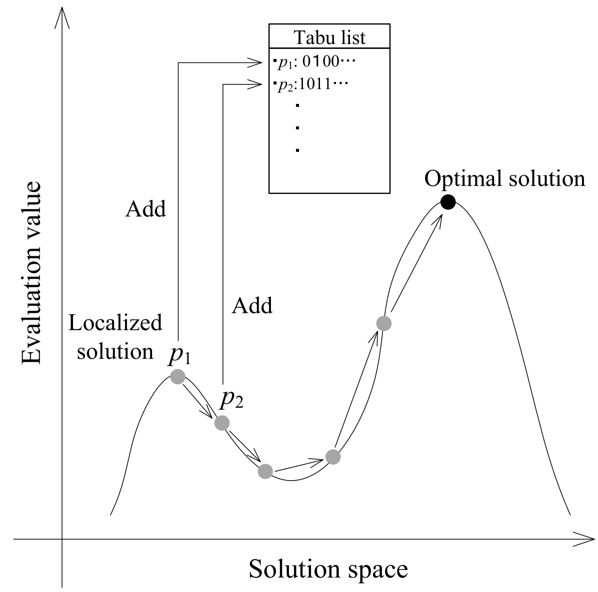


Fig. 1 TS solution retrieval.

Therefore, it is difficult for TS to fall into a local minimum (maximum). However, if the current candidate solution is a local optimum solution, TS may cycle and return to an original solution after moving from a current solution to another candidate solution. To solve this problem, TS maintains a tabu list of searched candidate solutions and prohibits moving to a candidate solution of the tabu list. However, if the tabu list holds searched candidate solutions for a long time period, the candidate solutions that can be moved are eliminated. Therefore, the tabu list elements are updated regularly. We use TS as an EC algorithm of IEC to test its potential for reducing the evaluation load of IEC users.

2.2 Interactive Tabu Search

ITS uses TS as an EC algorithm of IEC. Figure 2 shows the schematic of ITS. First, the ITS system randomly generates initial candidate solutions and presents them to the user. Next, the user chooses the favorite candidate solution from the presented candidate solutions; this candidate solution is added to the tabu list. Then, the system generates neighboring candidate solutions from the solution chosen by the user. Note that these neighboring candidate solutions are not included in the tabu list. Finally, the ITS system presents candidate solutions to the user again.

TS moves the current candidate solution from the neighboring candidate solutions to the current optimum candidate solution. Therefore, it is possible that the current candidate solution is worse than the optimum candidate solution from previous iterations. After the final evaluation, the ITS user chooses the favorite candidate solution from the candidate solutions in the tabu list.

An ITS user chooses his/her favorite candidate solution in each iteration. As a result, compared with NIGA, ITS has a greater likelihood of reducing user evaluation load. How-

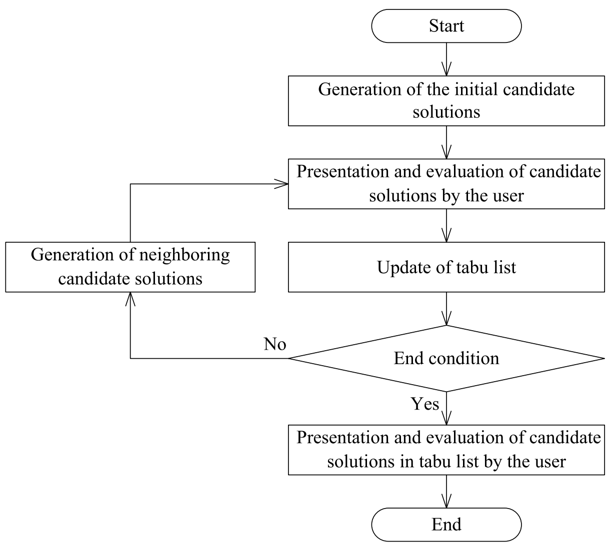


Fig. 2 Schematic of ITS.

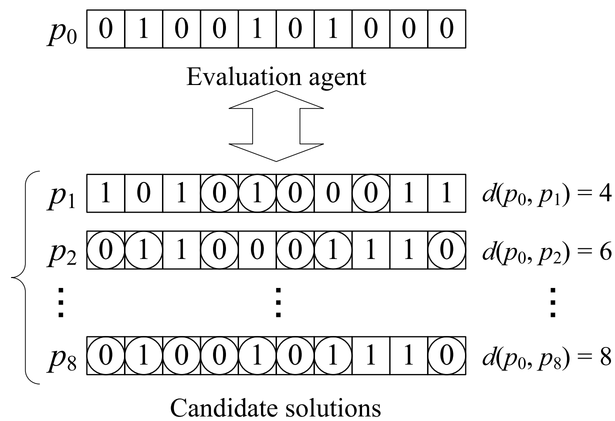


Fig. 3 Candidate solution evaluation with the evaluation agent.

ever, as mentioned previously, ITS candidate solutions converge locally because ITS is a local search method. Moreover, the necessity for the user to choose one candidate solution from presented candidate solutions even when a favorite candidate solution is not presented is problematic.

3. Performance Evaluation by Numerical Simulation

3.1 Candidate Evaluation with an Evaluation Agent

We examined the evolution performance of ITS and NIGA using a numerical simulation with an evaluation agent that imitated user preference (Kansei). The evaluation agent expresses a user's potential preferences and evaluates candidates using a similarity ratio (the Hamming distance). The evaluation agent is generated randomly. Figure 3 shows the method for evaluating candidates by the evaluation agent. When the similarity ratio between the evaluation agent p_0 and a candidate p_i is defined as $d(p_0, p_i)$, the absolute grading value of the candidate p_1 (p_2) is 4 (6). The value that rectifies the similarity ratio with the evaluation agent p_0 in a 10-stage evaluation is obtained using Eq. (1) as the evaluation value $fitness(p_i)$ of each candidate p_i.

$$fitness(p_i) = \frac{d(p_0, p_i)}{L} \times 10 \qquad (1)$$

Here, L is the gene length. In ITS, the evaluation agent chooses the candidate solution with the highest absolute grading value. If there are multiple candidate solutions with the highest absolute grading value, the evaluation agent determines one optimum candidate solution from the candidate solutions. On the other hand, NIGA uses absolute grading values for the evaluation values of each candidate solution.

The candidates evolve using the evaluation values assigned by the evaluation agent, because the bit string of the evaluation agent is not known by the IEC system in the simulation. When the candidates evolve well, the similarity ratio between the evaluation agent and candidate, which is optimized by the IEC system, becomes high whenever the IEC operation is repeated. Therefore, the simulation can quantitatively evaluate the evolution performance of each method.

Table 1 Parameters of ITS and NIGA.

	ITS	NIGA
Gene row	Bit string	
Gene length	10, 15, 20, 30, 40, 50 bits	
Candidate solutions	8	
Tabu list	8	–
Neighboring range	1 bit	–
Selection	–	Roulette selection + Elite preservation
Crossover	–	Uniform crossover
Mutation rate	–	3%

3.2 Simulation Result

It is necessary to set TS and GA parameters before comparing the evolution performance of ITS and NIGA so that each method shows the best performance. Table 1 shows the parameters, predetermined by the simulation, for both methods. The selection of the GA operation uses the normalized evaluation values in the range 0 ~ 100. A 15 bits simulation was performed because the running-shoes-design system, described in Sect. 4, has 15 bits. The neighboring range for ITS was 1 bit in the simulation and evaluation experiment, which in also described in Sect. 4. Therefore, ITS generates neighboring candidate solutions by flipping the current candidate solution within 1 bit. The simulation was performed using the parameters presented in Table 1. We performed 100 trials to remove the probabilistic influence.

Figure 4 shows the results of the performance comparison. The absolute grading value in Fig. 4 is the average of the absolute grading values for the highest scored candidates in each iteration. The absolute grading value for ITS is the

Table 2 Results of the *t*-test.

Gene length	Iteration number																				The number of significant
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	differences (2^{nd} – 20^{th})
10 bits	n.s.	**	**	**	**	**	**	**	**	n.s.	n.s.	**	**	**	**	**	**	**	**	**	17/19
15 bits	n.s.	*	**	**	**	**	**	**	**	**	**	**	**	**	n.s.	n.s.	n.s.	**	**	**	16/19
20 bits	n.s.	n.s.	**	**	**	**	**	**	**	**	**	**	**	**	**	**	n.s.	n.s.	n.s.	n.s.	14/19
30 bits	n.s.	n.s.	n.s.	n.s.	*	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	16/19
40 bits	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	**	**	**	**	**	**	**	**	**	**	**	**	**	**	14/19
50 bits	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	**	**	**	**	**	**	**	**	**	**	**	**	12/19

n.s. : Not significant, $*$: $p < 0.05$, $**$: $p < 0.01$

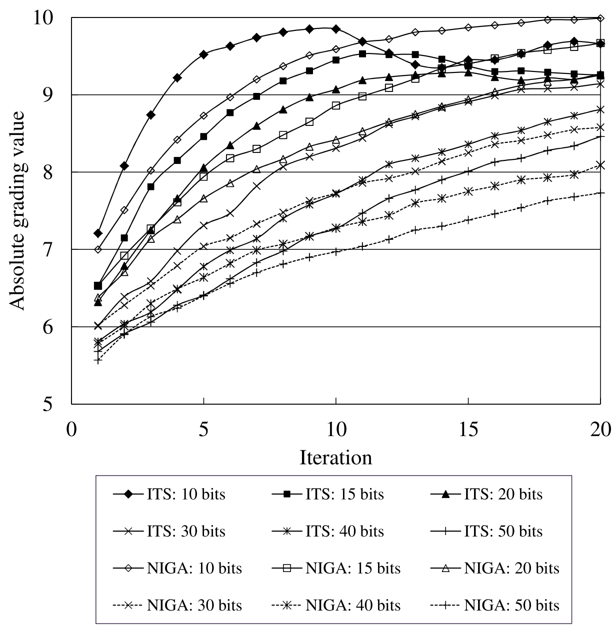


Fig. 4 Results of performance comparison.

highest absolute grading value for the candidate solutions in the tabu list in each iteration. The simulation results show that the evolution performance of ITS is higher than that of NIGA. When the gene lengths are 10, 15, and 20 bits, ITS shows a higher evolution performance than NIGA in the initial stages. When the gene lengths are 30, 40, and 50 bits, the evolution performances of ITS and NIGA are approximately equal in the initial stages. ITS shows a higher evolution performance than NIGA after the 10^{th} iteration because the solution space becomes complicated by the increase in gene length. As ITS candidate solutions converge locally owing to the local search, when the solution space is simple, i.e., the gene length is small, it is relatively easy for candidate solutions to converge to candidate solutions with higher evaluation values. However, in TS, when the solution space is complicated, i.e., the gene length is large, it becomes more difficult for candidate solutions to converge to candidate solutions with higher evaluation values.

Moreover, the evolution performance of ITS is delayed in the cases of gene lengths of 10 bits in the 10^{th} and 11^{th} iterations, 15 bits from the 15^{th} to 17^{th} iterations, and 20 bits from the 17^{th} to 20^{th} iterations. In the simulation, the tabu list size is eight. Consequently, the tabu list has a candidate solution with an evaluation value of 10; the absolute grading value of the ITS is 10 after eight iterations. However, if the number of iterations is greater than eight, the tabu list deletes a candidate solution with an evaluation value of 10 and the ITS performance deteriorates. Then, additional iterations are necessary until the optimum candidate solution is searched for again. During this operation, the evolution performance of ITS is delayed.

Therefore, we performed a *t*-test of the absolute grading value from the 2^{nd} to the 20^{th} iterations of ITS and NIGA. Table 2 shows the results of the *t*-test. The results of the *t*-test show that except in the case of large gene lengths in the initial stages, there is a statistically significant difference between the evolution performances of ITS and NIGA. The results in Table 2 and Fig. 4 are almost the same.

The simulation results show a statistically significant difference between the original algorithmic evolution performance of each method. However, an evaluation experiment with real users may not produce the same results as the simulation. Therefore, we conducted an evaluation experiment using real users to assess and compare the practical application of ITS.

4. Evaluation Experiment Using Test Subjects

4.1 Outline of the Experiments

We conducted an evaluation experiment using a running-shoes-design system to examine the effectiveness of ITS. Twenty-one university students, all in their 20 s, were the subjects in the experiment. After the experiment, the subjects evaluated their preferences for the generated candidates and the usability of the ITS and NIGA methods through a 5-stage evaluation. During the experiment, the order of the evaluation methods, ITS and NIGA, was randomly set for each subject. The subjects used a running-shoes-design system based on the concept that customers will select the running-shoes-design that they want to wear to enjoy running. The system test was terminated when 80% of the presented candidate solutions became satisfactory. Upon completion of each system test, the subjects selected their favorite can-

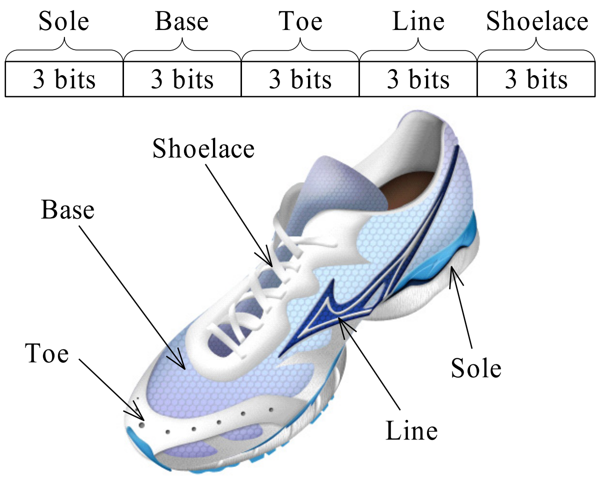


Fig. 5 Gene coding of the running-shoe-design system.

didate solution. In the NIGA system, the subjects selected the favorite candidate solution from the candidate solutions in the final iteration. In the ITS system, the subjects selected the favorite candidate solution from the candidate solutions in the tabu list. After the evaluation experiment, the subjects evaluated their satisfaction level with these two candidate solutions through a 5-stage evaluation. The evaluation experiment was performed using the parameters shown in Table 1.

An alternative experimental methodology could attempt to compare ITS and NIGA using the same time conditions and an equal time limit for both ITS and NIGA. However, we focused on whether a subject could generate a favorite candidate solution with the ITS and NIGA systems. Each subject required a different length of time to generate a favorite candidate solution; therefore, setting a uniform time limit was not possible. Moreover, ITS is a local search method, whereas NIGA is an area search method. ITS and NIGA are inherently different and can be assumed to require different length of time.

4.2 Running-Shoes-Design System

Figure 5 shows the gene coding of the running-shoes-design system. The running-shoes-design consists of five parts: sole, base, toe, line, and shoelace. Each part has eight designs, which are expressed by 3 bits gray coding. Therefore, the gene length is 15 bits. This system can generate 32,768 ($=2^{15}$) patterns. Figure 6 shows the running-shoes-design parts. The designs are generated by combining each part, as shown in Fig. 6.

4.3 Evaluation Interface

Figures 7 show the ITS and NIGA interfaces for the running-shoes-design systems.

During the ITS evaluation, the user chooses the favorite candidate solution from the presented candidate solutions by clicking the “Good” button displayed under each candi-

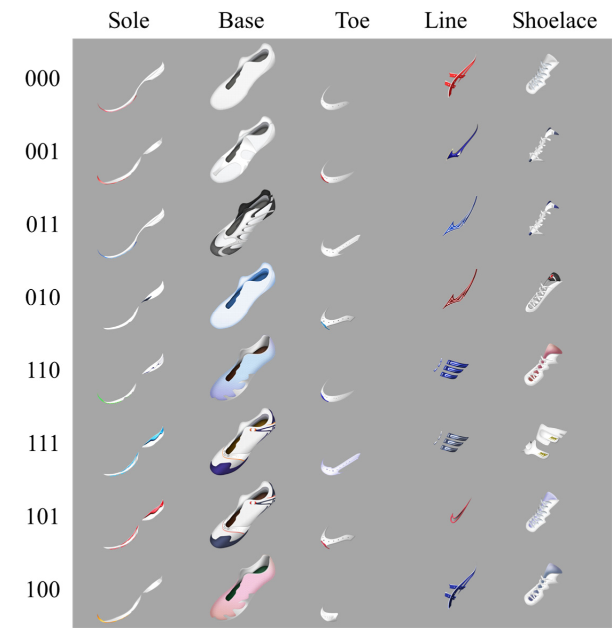


Fig. 6 Running-shoes-design parts.

(a) ITS

(b) NIGA

Fig. 7 Evaluation interfaces of the running-shoes-design system.

date. Then, when the user clicks the "OK" button to confirm a query that appears in a dialog window, new candidate solutions are generated and presented. When the user finishes creating candidates, the user clicks the "End" button and then chooses the favorite solution from the candidate solutions in the tabu list.

During the NIGA evaluation, the user inputs 10-stage evaluation values using a slider to select the evaluation values. When the user finishes evaluating all candidates, the user decides either to generate new candidates or finish creating candidates by clicking the "Next" or "End" button, respectively.

5. Experimental Result

5.1 Satisfaction Levels with Generated Candidates

In this section, we describe the satisfaction levels of users with the generated candidates and the number of iterations in each method.

Figure 8 shows the average and standard deviation of the satisfaction levels for candidates generated with each method. The satisfaction levels for ITS and NIGA were both approximately 4. Figure 9 shows the distribution of satisfaction levels. More than half of all subjects assigned a value of 4 to both ITS and NIGA. Moreover, we performed the Wilcoxon signed rank test to confirm the statistically significant differences between experimental results. Table 3 shows the results of the Wilcoxon signed rank test.

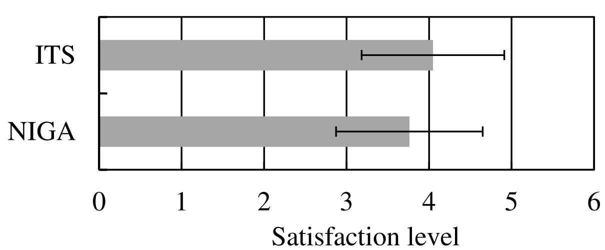


Fig. 8 Results of the satisfaction levels.

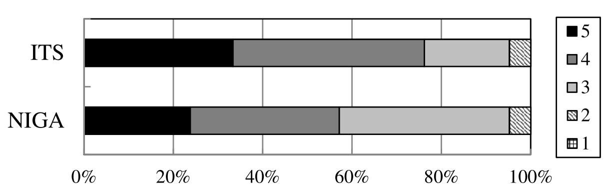


Fig. 9 Results of the subjects' distribution of the satisfaction levels.

Table 3 Results of the Wilcoxon signed rank test.

	Significance
Satisfaction level	n.s.
Iteration	$*$ NIGA
Usability	$**$ ITS
Evaluation time	$**$ ITS

n.s.: Not significant, $*$: $p < 0.05$, $**$: $p < 0.01$

The method listed in the significance column of Table 3 is the method with significantly better results. However, the evaluation experiments with real users did not confirm a statistically significant difference between the satisfaction levels for each method. Therefore, we confirmed that the two methods can generate candidates that satisfy the preferences (Kansei) of the subjects.

However, the number of iterations in each method resulted in different satisfactory candidates. Figure 10 shows the average and standard deviation of the iterations. There were a greater number of ITS iterations than NIGA iterations. Figure 11 shows the results of the subjects' distribution of the iterations. Approximately 75% of all subjects using ITS and most subjects using NIGA finished evaluating candidate solutions within the 10^{th} iteration. The Wilcoxon signed rank test results presented in Table 3 show the statistically significant differences between the number of iterations for each method when the significance level is 5%. These differences were caused by the neighboring convergence of ITS. If the presented ITS candidate solutions do not include the user's preferred candidates, ITS generates new candidate solutions that the user does not like. Then, ITS repeats the neighboring search and reaches a candidate solution that the user likes. As a result, there were a greater number of ITS iterations than NIGA iterations.

The simulation result, described in Sect. 3.2 for the number of iterations is different from the experimental results. In the simulation, there were a greater number of NIGA iterations than ITS iterations. This was owing to a subject attempting to create a more satisfactory candidate solution although the subject can create a satisfactory candidate solution. This can occur because ITS has a simpler and more intuitive evaluation interface than NIGA. As a result, ITS has a greater number of iterations than NIGA. Therefore, the statistically significant differences between the iterations of ITS and NIGA have been confirmed. However, this difference is only approximately 1.5 iterations.

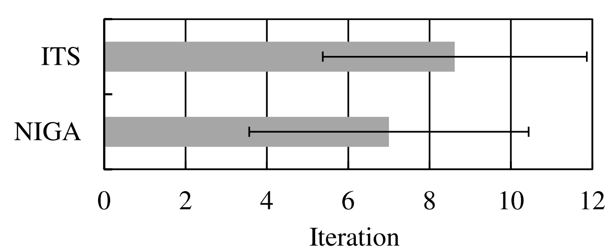


Fig. 10 Results of the iterations.

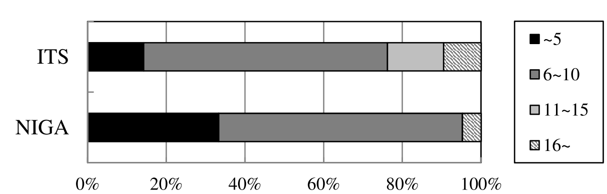


Fig. 11 Results of the subjects' distribution of the iteration.

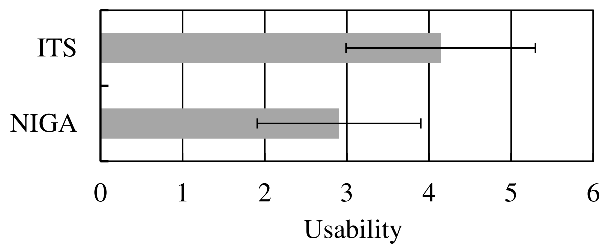


Fig. 12 Results for usability.

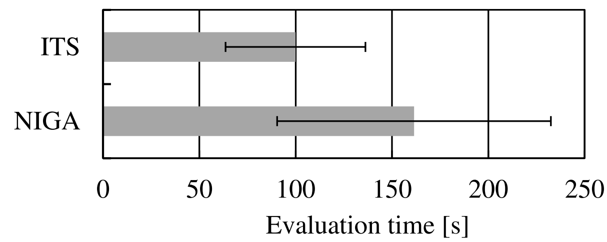


Fig. 14 Results of the evaluation time.

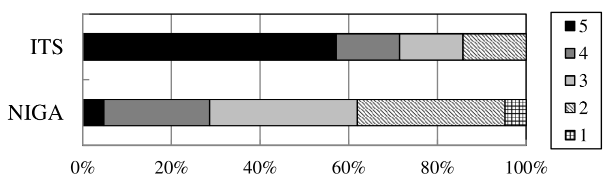


Fig. 13 Results of the subjects' distribution of usability.

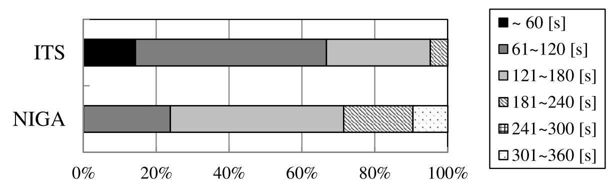


Fig. 15 Results of the subjects' distribution of evaluation time.

5.2 Usability and Evaluation Time of Each System

In this section, we describe the usability and evaluation time of each method.

Figure 12 shows the average and standard deviation of the usability of each method. The usability of ITS was rated higher than that of NIGA. Figure 13 shows the distribution of the usability. Approximately 75% of all ITS subjects and approximately 60% of all NIGA subjects rated the usability of both methods as 4. The Wilcoxon signed rank test results in Table 3 shows the statistically significant differences between the usability of each method when the significance level is 1%. It can be concluded that the ITS interface is easier to use for evaluating candidates compared with the NIGA interface; thus, ITS is a simpler evaluation method than NIGA.

Reducing the time spent to evaluate candidates can decrease the overall user evaluation load in an IEC system. The evaluation time for each system is the time from initial presentation of candidate solutions to the selection of the favorite candidate solution from the candidate solutions in the tabu list (ITS) or the final iteration (NIGA). Figure 14 shows the average and standard deviation of the time spent to evaluate candidates using each method. The evaluation times for ITS were shorter than those for NIGA. Figure 15 shows the results of the subjects' distribution of the evaluation times. For ITS, approximately 70% of all subjects finished evaluating candidate solutions within 120 [s]. However, for NIGA, approximately 70% of all subjects spent more than 120 [s] evaluating candidate solutions. The Wilcoxon signed rank test results in Table 3 show the statistically significant differences between the evaluation times of each method when the significance level is 1%. ITS had a relatively short evaluation time; however, ITS required a greater number of iterations than NIGA. In ITS, the subjects choose the favorite candidate solution from the presented candidate solutions in every iteration. However, in NIGA, which has a 10-stage evaluation process, the subject must input an evaluation value for each candidate solutions in every iteration. Therefore, although ITS requires a greater number of iterations than NIGA, ITS has a shorter evaluation time.

The usability of the ITS interface was ranked higher than that of the NIGA interface and the evaluation times for ITS were shorter than those for NIGA. Therefore, it is possible that ITS is effective for reducing the overall evaluation load of IEC users.

5.3 Subjects' Comments

We collected subjects' comments regarding ITS evaluation and recommendation for improvements. The most significant comment can be summarized as follows.

- ITS requires more time to obtain preferred candidate solutions.

This comment is predictable because ITS performs a local search. ITS forces the subject to choose a candidate solution even when the subject's preferred candidate solution is not among the possible selections. To solve this problem, it is necessary to apply a parallel TS algorithm. For example, the user divides candidate solutions into three groups; "like," "not bad," and "do not like." The ITS system performs parallel searches for information on the basis of the user identified groups of candidate solutions. Moreover, applying a hybrid GA and TS is effective for reducing the amount of time required to obtain preferred candidate solutions.

6. Conclusion

We showed the effectiveness of ITS for reducing the evaluation load of IEC users. The numerical simulation results confirmed a statistically significant difference between the evolution performance of NIGA and ITS. However, the results of evaluation experiments with real users did not confirm this result. Nevertheless, the statistically significant difference in the usability of each method was confirmed. Therefore, it is possible that ITS is effective in reducing the evaluation load of IEC users. In future studies, we will consider a parallel TS algorithm to improve a local search of ITS.

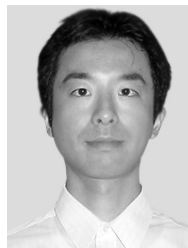
Acknowledgment

This work was supported by MEXT.KAKENHI (24500264).

References

[1] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," Proc. IEEE, vol.89, no.9, pp.1275–1296, 2001.
[2] S.B. Cho and J.Y. Lee, "A human-oriented image retrieval system using interactive genetic algorithm," IEEE Trans. Syst. Man Cybern. A, Syst. Humans, vol.32, no.3, pp.452–458, 2002.
[3] H.S. Kim and S.B. Cho, "Fashion design using interactive genetic algorithm with knowledge-based encoding," Yaochu Jin, Knowledge Incorporation in Evolutionary Computation, pp.411–434, 2005.
[4] Y. Bamba, J. Kotani, and M. Hagiwara, "An interior layout support system with interactive evolutionary computation using evaluation agents," Joint 5th Int. Conf. on Soft Computing and Intelligent Systems and 11th Int. Symp. on Advanced Intelligent Systems (SCIS&ISIS2004), WE-2-4, pp.1–6, 2004.
[5] M. Miki, H. Orita, and T. Hiroyasu, "Design of sign sounds using interactive genetic algorithm," Proc. IEEE Int. Conf. Systems, Man, and Cybernetics, pp.93–98, 2006.
[6] H. Takagi and M. Ohsaki, "Interactive evolutionary computation-based hearing aid fitting," IEEE Trans. Evol. Comput., vol.11, no.3, pp.414–427, 2007.
[7] H. Inoue and H. Miyagoshi, "Behavior evolution of pet robots with human interaction," Proc. 2nd Int. Conf. on Innovative Computing, Information and Control, pp.23–26, 2007.
[8] Y. Ishino and T. Terao, "Marketing data analysis using simulated breeding and inductive learning techniques," Trans. of the Japanese Society for Artificial Intelligence, vol.12, no.1, pp.121–131, 1997 (in Japanese).
[9] B. Johanson, Automated Fitness Raters for the GP-Music System, University of Birmingham, Masters Degree Final Project, 1997.
[10] I.S. Lim and D. Thalmann, "Tournament-style selection for browsing temporal signals," ACM Symp. on Applied Computing, pp.570–573, 2000.
[11] P.J. Angeline and J.B. Pollack, "Competitive environments evolve better solutions for complex tasks," 5th Int. Conf. on Genetic Algorithms, pp.264–270, 1993.
[12] Y. Watanabe, T. Yoshikawa, and T. Furuhashi, "A study on application of fitness inference method to PC-IGA," 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp.1450–1455, 2007.
[13] F. Glover, "Tabu search - Part I," ORSA Journal on Computing, vol.1, no.3, pp.190–206, 1989.
[14] G.W. Klau, N. Lesh, J. Marks, and M. Mitzenmacher, "Human-guided tabu search," Proc. 18th National Conf. on Artificial Intelligence, pp.41–47, 2002.
[15] S. Munemoto, N. Katoh, and G. Imamura, "Finding an optical floor layout based on an orthogonal graph drawing algorithm: Interactive multi-objective optimization using tabu search," J. Architecture, Planning and Environmental Engineering, no.529, pp.279–286, 2000 (in Japanese).

Hiroshi Takenouchi is a doctoral student member of the Graduate School of Kansai University in Osaka, Japan. He received B.E. and M.E. degrees in Engineering from Kansai University in 2008 and 2010, respectively. His current research interests include Interactive Evolutionary Computation using multiple person's Kansei evaluation and the Interactive Evolutionary Computation interface.



Masataka Tokumaru is an Associate Professor at Faculty of Engineering Science at Kansai University in Osaka, Japan since 2008. He received B.E. and M.E. degrees in Engineering from Kansai University in 1994–1997. During 1997-2007, he had been a Research Associate and a Lecturer of Department of Electronics at Kansai University. He received a Dr. (Eng.) degree from Osaka City University, Japan in 2001. His research interests are Kansei information processing and evolutionary computation. He is a member of the Japan Society of Kansei Engineering, the Japan Society for Fuzzy Theory and Intelligent Informatics, and Human Interface Society.

Noriaki Muranaka is a Professor of Faculty of Engineering Science at Kansai University in Osaka, Japan since 2005. He received B.E. and M.E. degrees from Kansai University, Japan in 1972 and 1974, respectively, and a D.E degree from Himeji Institute of Technology, Japan in 1990. He is involved in research on multivalued logic circuit, intelligence, Kansei and intention information processing, safety and disaster prevention information processing. Currently, his research interests include brain information processing and welfare technology. He is a member of the Institute of Electrical Engineers of Japan, the Japan Society of Kansei Engineering, and IEEE, among others.