An Intuitive and Traceable Human-based Evolutionary Computation System for Solving Problems in Human Organizations

Kei Ohnishi
Kyushu Institute of Technology
Iizuka, Fukuoka, Japan
ohnishi@cse.kyutech.ac.jp

Tomohiro Yoshikawa
Nagoya University
Nagoya, Aichi, Japan
yoshikawa@cse.nagoya-u.ac.jp

Tian-Li Yu
National Taiwan University
Taipei, Taiwan
tianliyu@ntu.edu.tw

ABSTRACT

Human-based evolutionary computation (EC), for which people act as executors of all evolutionary operators, can be used to solve problems in human organizations. We previously developed a human-based EC system that represents solutions as tags (words) and allows people to evaluate solutions by clicking corresponding tags. Although the system was easy and intuitive to use, it could not handle problems for which solutions are represented as long sentences. In addition, the system could not trace the evolution of solutions. Traceability is a must for the system to be widely and reliably used. In this study, we thus develop a human-based EC system that allows solutions to be represented as both sentences and tags. A function for tracing the evolution of solutions is embedded into the system. The function asks a solution creator to specify which existing solutions influenced the solution creation. We conduct an experiment in which 18 human subjects use the system and then fill out a survey. The results show that the system creates better solutions than those created by each human subject independently. Furthermore, the evolution tree generated from the information given by solution creators is used to confirm that the system allows the evolution of solutions to be traced.

CCS CONCEPTS

• **Human-centered computing → Collaborative content creation**; Social tagging systems; • **Computing methodologies →** *Self-organization*;

ACM Reference Format:
Kei Ohnishi, Tomohiro Yoshikawa, and Tian-Li Yu. 2019. An Intuitive and Traceable Human-based Evolutionary Computation System for Solving Problems in Human Organizations. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19).* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Evolutionary computation (EC) is a framework of optimization methods that model genetics and evolution. Many algorithms based on the concept of EC have been developed. There are two main steps in EC, namely selection, which mimics natural selection, and operations, which mimic crossover and mutation. These steps can also be viewed as the actions of agents, and thus the entire EC can be viewed as a multi-agent system [5]. In human-based EC, human agents perform the selection as well as the operations [5]. Since human-based EC has humans produce and evaluate solutions, it can solve problems in human organizations for which only humans can evaluate the quality of solutions. Human-based EC can be applied to complex problems such as global warming.

Human-based EC systems can be either centralized or decentralized. In centralized human-based EC systems [3], the system manages all solutions created by humans and shares them among humans in a central location, such as a web page. In decentralized human-based EC systems [4, 8], humans manage their own solutions and share them with other humans in a local area through direct connections. Solutions can be shared over a mobile ad-hoc network formed by wireless communication devices.

In our previous study, we developed a human-based EC system that uses a tag cloud as a way to discuss how to utilize the huge amount and variety of data available on the Internet [3]. A tag cloud is a way to visualize tags assigned to web site content, where the size of a tag is related to the tag frequency or significance. We developed a web-based system for discussing suitable tags to be assigned to various images. The system was a centralized human-based EC system. We evaluated the developed system using subjective tests. Although the task in that study was not to discuss how to utilize the data but rather to determine suitable tags, the functionality that enables us to discuss how to utilize the data exists. We can easily realize the desired functionality by replacing the discussion of tags with a discussion of ideas for utilizing the data.

The human-based EC system developed in our previous study displays at most ten tags, which are displayed in the order in which they were created. Additional tags are not displayed in the tag cloud. In the first-level evaluation of the system through subjective tests, this did not cause any problems because the number of tags created was ten or fewer. However, to improve the quality of the tags, it is necessary to create a mechanism that can store all of the tags created and then selectively display the tags preferred by many evaluators. We thus proposed a generation gap model that allows all tags with sufficient fitness values to remain in the tag cloud, which enables all tags created to be evaluated by people [1]. Tags created after the upper limit of the number of tags that can be displayed has been reached are stored in a queue in

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the order they were created, and at the conclusion of each generation, a certain number (a previously determined constant) of the lower-fitness tags in the tag cloud are replaced by the corresponding number of tags in the queue. The fitness values of all tags in the cloud are then reset.

A problem with the human-based EC system with the generation gap model [1] is that all solutions for a given problem have to be represented as tags. For problems for which solutions should be explained in natural language, the use of only tags would not accurately represent the intentions of the solution creators. For the system to be widely used for solving problems in human organizations, it should allow more flexible representations of solutions.

Moreover, the system did not have a function for tracing the evolution of solutions. Since human-based EC systems are complex systems in which people interact with each other non-linearly, it is not easy for people to forecast and analyze the problem solving process. However, for human-based EC systems to be widely and reliably used, traceability is a must.

In the present study, we modify the above human-based EC system to allow it to represent solutions as headlines and details. Tags are used to represent the headlines of solutions and a tag cloud is used to display the tags. This representation allows people to easily grasp the general view of all solutions. Sentences are used to represent the details of solutions. These sentences, which can be appended, are displayed by the system when a participant selects the corresponding tags. In addition, we create a function for tracing the evolution of solutions. For automatic traceability, the system needs to detect which existing solutions influenced new solutions. However, this is impossible. Our idea is for the system to ask a solution creator to specify which existing solutions influenced their solution right after solution creation. This allows the evolution of solutions to be traced.

In the present study, we also conduct an experiment in which human subjects use the system for problem solving and then fill out a survey. Then, we apply statistical hypothesis testing to the survey data and show that the system creates better solutions than those created by each subject independently. Furthermore, we build an evolution tree using the information obtained from solution creators and use it to confirm that the evolution of solutions is indeed traceable.

The remainder of this paper is organized as follows. In Section 2, we briefly review related research. In Section 3, we describe the proposed human-based EC system. Section 4 shows the experimental results. Finally, Section 5 presents conclusions and suggests areas for future research.

2 RELATED WORK

The human-based genetic algorithm (GA) [5], a type of human-based EC, was first applied to problems for which the problem itself and its solutions must be described in natural language [5, 6]. In this application, the human-based GA used a centralized online message board to manage communication among human agents. The applications of the human-based GA are not limited to those that involve natural language. One study [2] compared an interactive GA and a human-based GA in terms of their ability to solve problems not described in natural language. It was shown that crossover and mutation operations performed by human agents are useful for solving such problems.

The GA framework has been utilized for modeling creative problem solving processes in human organizations [6, 7]. In [6], the components and procedures in human organizations were considered to be genes, individuals, population, selection, crossover, and mutation in a GA. In [7], a data mining technique was applied to discussions by people on an online message board to find solutions to a given problem. The words important for solving the problem, considered to be building blocks in a GA, were extracted and then fed back to the people in real time. This procedure was iterated to produce better solutions by combining the obtained building blocks, which is considered to be crossover in a GA.

As mentioned in Section 1, we previously developed a centralized human-based EC system for creating ideas for utilizing a variety of data on the Internet, in which a tag cloud is used to share solutions among participants [3]. We also proposed a method for sharing solutions in a decentralized human-based EC system [4], where the participants represent their interest as a numerical vector. The distances between the vectors of participants are used for regulating the propagation of solutions among participants. We also examined the relationship between the timing of sharing solutions with others and search performance in a decentralized human-based EC system [8].

The creative activities of humans have been modeled as EC in several studies [9–11].

In [9], the authors chose the design of an advertisement to be published in a social network service as the task for a human-based EC system and then compared two types of human-based EC system with a method in which each person independently thinks of a design. One of the human-based EC systems relied only on crossover for the creation of new solutions (i.e., new solutions were created by combining two existing solutions). The other system relied only on mutation (i.e., new solutions were created by modifying an existing solution). The designs created by the three methods were subjectively evaluated in terms of divergence, relevance, and effectiveness by people who had not participated in the problem solving.

In the present study, we also compare solutions created by a human-based EC system with ones created by people alone in advance of the problem solving using the system. However, the people who conducted the comparison also participated in the problem solving. People were allowed to freely choose how they create a solution.

In [11], the authors proposed a general procedure to realize a crowd-sourcing-based design and allowed human-based EC as an option in a step of the procedure. In addition, they first considered a generative design task, where humans provide design purposes and constraints and then computers generate a variety of designs meeting these purposes and constraints, for crowd-sourcing according to the proposed procedure. Then, they proposed a crowd-sourcing-based method to reasonably evaluate the quality of the created designs. In the evaluation method, people who had participated in the crowd-sourcing evaluation first created multiple criteria for evaluating quality and then ranked the designs based on these criteria. The method was shown to give rankings closer to those of experts compared to a method in which people freely ranked the designs. This method is applicable to human-based EC, which relies on human subjective evaluations of solutions. However, in our human-based EC system, we allow people who participate in the problem solving to freely evaluate solutions. Our main concern regarding evaluations is whether people can intuitively and easily evaluate products.

In [10], the authors conducted simulations of human-based EC in which various humans were modeled in terms of fitness functions and behaviors in solution evaluation and creation, and then made three hypotheses on the relationship between the characteristics of a human group that executed the human-based EC and problem solving performance. Next, they conducted experiments to validate the three hypotheses. The hypotheses were roughly (1) a homogeneous human group in terms of the degree of problem understanding yields better performance, (2) a well-balance human group in terms of the ratio of people who mainly try to create solutions and who mainly try to evaluate solutions yields better performance, and (3) the use of various evolutionary operators yields better performance.

In the experiment conducted in the present study, the participants were university students from the same department, and were thus mostly heterogeneous. We did not attempt to find a balance of people with different behaviors. Nevertheless, the experimental results show that the group created a nearly constant number of solutions and evaluated solutions a nearly constant number of times in each generation. This means that there was some balance even though we cannot quantify it. Regarding the diversity of evolutionary operators, we observed from the trace of the evolution of solutions that the number of existing solutions that influence solution creation depended on the individual. Therefore, the participants used a variety of evolutionary operators.

3 PROPOSED HUMAN-BASED EC SYSTEM

The proposed human-based EC system is built upon our previously developed system [1]. The following subsections describe the system and explain how it is different from its predecessor. The proposed system was developed with PHP and HTML, whereas the previous one was developed with Java and HTML.

3.1 General Features of Human-based EC

In human-based EC, humans execute all evolutionary operators. The differences between human-based and standard EC are summarized in Table 1.

For people to widely use human-based EC systems, graphical user interfaces that allow people to intuitively execute EC must be developed. Because people use different evaluation criteria and their characteristics vary over time, it is hard for people, especially experts on EC, to trust human-based EC systems. Therefore, to increase trust in such systems, we need to enhance solution traceability. Lastly, incentives are required for people to contribute to human-based EC systems.

The present study develops an intuitive graphical user interface and enhances solution traceability. The problem of providing incentives for contributions in a human-based EC system will be considered in future work.

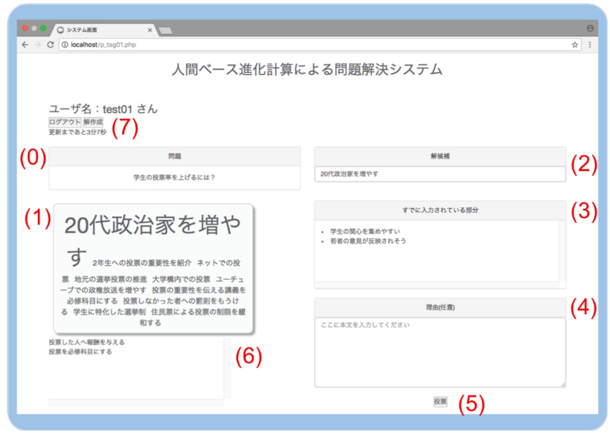

Figure 1: Main window of proposed human-based EC system. Labels (0) to (7) are used for explanation and are not included in the actual system.

3.2 Representation of Solutions

For problems that occur in human organizations, some solutions can be represented by a few words whereas others require long sentences. Therefore, to apply a human-based EC system to a wide variety of problems, we adopt two representations for solutions, namely tags and sentences. A tag is a summary of a solution. Tags are displayed in a tag cloud. Sentences, which were not used in our previous human-based EC system, provide the details of a solution. When an appropriate representation of a solution is a tag, sentences can be used to explain the reason the solution was chosen.

The representation of a solution using a tag is the same as in our previous system. The new part is the use of sentences together with tags.

3.3 Interface for Displaying and Evaluating Solutions

The interface for displaying and evaluating solutions in the proposed system is basically the same as that in our previous system. The interface for representing a solution as sentences and that for evaluating a solution by clicking a vote button are developed here (in our previous system, a solution was evaluated by clicking a tag that represented the solution).

A solution to a problem is represented by a tag and sentences, as mentioned in Section 3.2. Solutions represented by tags are displayed in a tag cloud. Figure 1 shows the main window of the system developed in this study. The widget labeled (0) displays the problem to be solved and that labeled (1) is the tag cloud.

When a solution represented by a tag in the tag cloud is clicked, the clicked tag is displayed by the widget labeled (2) and a solution represented by sentences is displayed by the widget labeled (3). New sentences can be added to the solution using the widget labeled (4) by any user. In general, new sentences are added to an existing solution as a supplementary explanation.

Table 1: Differences between human-based and standard EC.

		Standard EC	Human-based EC
(1)	Kinds of agent	Computer(s)	Humans
(2)	Evaluation criteria for agents	Identical	Different
(3)	Methods for producing solutions by agents	Identical	Different
(4)	Time variation of agent characteristics	No	Yes
(5)	Incentives for contributions by agents	Not necessary	Necessary

To evaluate a clicked tag as being good, the user can click the vote button, which is the widget labeled (5). The fitness value of the clicked tag is then increased by one. A user who participates in the problem solving can evaluate a given solution just once in a generation. The fitness value of every solution is set to zero at the beginning of a generation. A larger fitness value indicates a better solution. When the vote button is clicked, new sentences, if any, are added to the solution.

The font size of a tag in the tag cloud is dynamically changed based on the tag's fitness value. To determine the font size of a tag, we first transform the fitness value, f, to a relative fitness value, F, according to Equation (1).

$$F = \frac{f - f_{min}}{f_{max} - f_{min}}, \tag{1}$$

where f_{max} is the current largest (best) fitness value among all tags and f_{min} is the smallest (worst) current fitness value. The font size for displaying the tag is then based on F as follows. When $0.0 \leqq F \leqq 0.2$, the font size is 16 points. When $0.2 < F_n \leqq 0.4$, the font size is 18 points. When $0.4 < F \leqq 0.6$, the font size is 26 points. When $0.6 < F \leqq 0.8$, the font size is 32 points. When $0.8 < F \leqq 1.0$, the font size is 48 points. However, if $f_{max} = f_{min}$, which includes the situation where the fitness values of all tags are set to zero at the beginning of a generation, all tags are displayed using a font size of 26 points.

The maximum number of tags displayed in the tag cloud is 10.

3.4 Interface for Displaying Solutions in a Queue

This component was developed in the present study.

As mentioned in Section 3.3, the interface for displaying tags is the tag cloud and the maximum number of tags in the tag cloud is 10. Tags created when the tag cloud is already full are placed in a queue in the order they were created. However, since we assign a fitness value of one to a solution created during a generation, if there are tags with a fitness value of zero in the tag cloud, one of them is replaced by the created tag (its fitness value is one). The removed tag, whose fitness value is zero, gets placed at the start of the queue. The tags in the queue are displayed by the widget labeled (6) in Figure 1. Thus, participants can see all tags in the queue. We can expect that these tags also give ideas to the participants. However, participants cannot see the sentences corresponding to a tag in the queue.

We can evaluate tags in the queue during a generation. To do this, we first input the name of the target tag in the queue using the widget labeled (2) and then click the vote button. The fitness value of the target tag is then increased by one. Then, if there are tags with a fitness value less than that of the target tag in the tag cloud, one of them is replaced by the target tag and the removed tag from the tag cloud gets placed at the start of the queue.

A queue was used for the same purpose in our previous system, but the tags in the queue were not displayed.

3.5 Interface for Creating Solutions

This component was developed in the present study. To obtain the information required for tracing the evolution of solutions from participants, which is described in Section 3.6, the interface for creating solutions is displayed in an independent window. In contrast, in our previous system, solutions were created in the main window (this window displayed solutions and allowed solutions to be evaluated and created).

Specifically, when the "Create solution" button (widget (7) in Figure 1) is clicked, a new window is shown for creating solutions, as shown in Figure 2. The interfaces for inputting a solution as a tag and sentences are labeled (8) and (9) in the new window, respectively. The widget labeled (10) is used for selecting any number of existing solutions that influenced the present solution creation. We explain this in Section 3.6. The input solution is confirmed by clicking the "Creation" button. Other solutions can be created at any time in this window. To return to the main window shown in Figure 1, the user clicks the "Return" button (widget (12)). The created solution can be only a tag (sentences are not required).

As mentioned in Section 3.4, a newly created solution is assigned a fitness value of one at the time of creation, so that if there are any solutions with a fitness value of zero in the tag cloud at the moment of creation, one of them is replaced by the newly created solution. The removed tag gets placed at the start of the queue.

According to the concept of human-based EC, solution creation by participants is regarded as the use of operations (crossover and mutation) in EC.

3.6 Interface for Obtaining Information for Tracing the Evolution of Solutions

As mentioned in Section 3.5, this component was developed in the present study.

The widget labeled (10) in the window for solution creation shown in Figure 2 is used to select an arbitrary number of present solutions (i.e., tags currently in the tag cloud) that influenced the present solution creation.

3.7 Generation Gap Model

The generation gap model used in the developed system is the same as that used in our previous system [1].

Figure 2: Window used for solution creation. The labels (8) to (12) are used for explanation and are not included in the actual system.

The generation gap model is introduced to give all created solutions equal opportunity to be evaluated and to lower the dependency of the survival probability of a solution on the timing of its creation (i.e., solutions created earlier are more likely to survive). In the generation gap model, the mechanism used to replace tags in the tag cloud by tags in the queue tries to give equal opportunity of evaluation. In addition, the mechanism used to reset the fitness values of all solutions created so far at every generation tries to remove the dependency on creation timing. The procedure of the generation gap model is as follows.

(1) When the upper limit of the number of tags that can be displayed is reached, any additional tags are stored in the order in which they are created (i.e., they are stored in a queue).
(2) A fixed period of time (for example, 10 minutes) is considered to be one generation.
(3) When the present generation becomes the next generation, a fixed percentage, X%, of the displayed tags with the lowest fitness values are replaced by the same number of tags from the queue in the order they were created. In the proposed system, we set X to 50%. If there is an insufficient number of tags in the queue to replace 50% of the tags in the cloud, the number of tags replaced is the number of tags in the queue.
(4) At the beginning of each new generation, the fitness of all tags in the tag cloud is set to zero.

4 EXPERIMENT

4.1 Purpose and Conditions

The experiment was conducted to examine the effectiveness of the developed human-based EC system by applying statistical hypothesis testing to the survey data obtained after problem solving by the system. It was also used to determine whether the introduced function enables us to trace the evolution of solutions.

The number of human subjects who participated in the experiment was 18 (3 graduate students and 15 undergraduate students from the university of the first author). The parameter values of the system are shown in Table 2. The maximum number of tags displayed in the tag cloud and the percentage of tags in the tag cloud that are replaced by tags in the queue were 10 and 50%, respectively, but those are originally the system parameters.

Table 2: Parameter values of the human-based EC system used in the experiment.

Parameter	Value
Number of generations	12
Duration of one generation (minutes)	5
Maximum number of tags displayed in the tag cloud	10
Percentage of tags in the tag cloud replaced by tags in the queue (%)	50

The problem that the human subjects had to solve using the system was "What is the most necessary thing for a happy life?". For the problem, they were asked to use tags to represent the solutions themselves and sentences to represent reasons for choosing the solutions.

Before the experiment, all human subjects were told how to operate the human-based EC system, the purpose of the experiment, and the time needed for the experiment. Then, they were asked to memorize their own best solution (i.e., created alone) to the given problem. The personal best solution was compared with the best solutions created using the system in the survey given after the problem solving. Then, the human subjects were allowed to go anywhere they could connect to the Internet. The system was implemented on a Web server with a global IP address, so it was available on the Internet.

We also asked the subjects the following four questions after the problem solving using the system.

(1) Did the system create better solutions than your own best solution? Please answer yes or no.
(2) Was the problem solving process by the human subjects creative? Please answer yes or no.
(3) Was the system easy to operate? Please answer yes or no.
(4) Please write your feelings freely.

4.2 Results

4.2.1 Behaviors of Human Subjects. The numbers of solutions created and solution evaluations over time are shown in Figure 3. The figure shows that the number of solutions created was slightly more than average in the first generation and slightly less than average in the final generation, but almost constant in the other generations, and that the number of solution evaluations was slightly less than average in the first generation and slightly more than average in the final generation, but almost constant in the other generations. For the given problem, it was easy to keep coming up with new ideas. In addition, the human subjects were motivated to contribute to the problem solving (creating and evaluating solutions) the entire time. We do not have enough data to judge the quality of the contributions. However, some human subjects stated in the survey that there were many nonsense solutions.

4.2.2 Statistical Hypothesis Testing. The results for questions 1 to 3 shown above are shown in Table 3. Summaries of the answers to question 4 are shown in Table 4 (meaning was preserved).

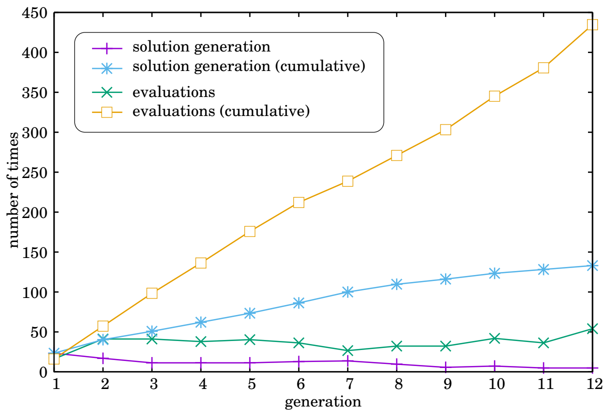

Figure 3: Numbers of solutions created and solution evaluations over time.

Table 3: Results for questions 1 to 3.

	Yes	No
question 1	15	3
question 2	16	2
question 3	18	0

The sign test with a significance level of 1 % was applied to the answer data to question 1. It confirmed that there is a statistically significant difference between the proposed system and individual solution creation. This means that the developed human-based EC system can create better solutions than those independently created by the human subjects. The proposed system is thus effective, at least for the given problem.

Regarding the answers to questions 2 and 3, since "neither yes nor no" was not an option, we cannot judge whether there is a statistically significant difference between the proposed system and individual solution creation. However, the answers suggest that the system yielded more creative solutions than those created by the human subjects alone and that the usability of the system was quite high.

The answers to question 4 in Table 4 indicate that there were many solutions that were unsuitable for the problem. The reason for this might be that many of the human subjects had known each other and were thus likely to joke around. Some human subjects did not mind this but others claimed that the system should have allowed them to remove unsuitable tags by voting. In standard EC, the system can select better solutions to remove unnecessary solutions, but when there are many unnecessary solutions, selecting better solutions does not work effectively. We need to implement a function that will remove unnecessary solutions by voting. The functions for selecting necessary solutions and removing unnecessary ones can be selected according to results of real-time evaluations of the quality of the entire process of problem solving.

We also recognize the necessity of utilizing the information in the queue, as mentioned by a user (see Table 4). The user might have thought that the maximum number of tags displayed in the

Table 4: Summeries of answers to question 4.

Feelings regarding problem solving using the system
There were many nonsense solutions. (2 cases)
Some solutions were never displayed in the tag cloud. (1 case)
I did not feel like evaluating a given tag two or more times. (1 case)
Similar solutions seemed to stay in the tag cloud for a long time. (1 case)
Opinions regarding the environment in which the system was used
Human subjects should avoid meeting face to face during the problem solving. (1 case)
Opinions regarding usability and functionality of the system
The system should allow unsuitable solutions to be removed by clicking a button. (5 cases)
The system should allow us to update the display of the tag cloud at any time by clicking a button. (3 cases)
The system should allow us to see the reasons for the solutions in the queue. (2 cases) It was a little difficult to push the "Return" button on a smart phone because it was very close to the "Create solution" button. (1 case)
The system should display the generation number. (1 case)
I wanted to select some solutions in the queue that influenced my solution creation. (1 case)
I wanted the system to visualize the influence relationship among solutions. (1 case)
The system should change the color of fonts for displaying tags. (1 case)
Opinions regarding problems that the system can solve
Philosophical problems are unsuitable for the system. (1 case)
Problems for which concrete and practical solutions are easy to come up with are suitable for the system. (1 case)

tag cloud (10) was too small. Additionally, five (half) of the tags in the tag cloud were replaced by five tags in the queue at the beginning of each generation, so the number of spaces in the tag cloud for better solutions that the human subjects supported was just five. Furthermore, some human subjects might have forgotten that they could evaluate tags in the queue and display them in the tag cloud during a generation. We need to implement a function that dynamically changes the maximum number of tags displayed in the tag cloud according to results of real-time evaluations of the quality of the entire process of problem solving. This might enhance the satisfaction of participants.

As shown in Table 4, the subjects stated that suitable problems for the system are those for which people can easily come up with concrete and practical solutions. For the problem used in the study, "What is the most necessary thing for a happy life?", the human subjects provided a variety of solutions and determined which solutions they felt the strongest about. Various ideas could be contributed for solving the given problem without specialized knowledge. The solutions were kept being created, as shown in Figure 3. For problems that require specialized knowledge to be solved, the performance of the solution search will depend on the knowledge of participants. As presented in [10], a homogeneous group of people, in terms of the degree of problem understanding, are better at problem solving. Therefore, we should assign a group of appropriate experts to a specialized problem. In this case, solutions would be selected based mainly not on sympathy but on rationality. This should be considered when creating benchmark problems for human-based EC in future work.

4.2.3 Evolution of Solutions. Figure 4 shows the evolution of solutions up to those in the tag cloud in the final generation. A list of solutions related to the final ones, namely target solutions to which crossover and mutation operators were applied, is shown in Table 5. These solutions were originally stated in Japanese. The rectangles in Figure 4 stand for solutions and the numbers in the rectangles correspond to the numbers shown in Table 5. The numbers indicate the order of solution creation.

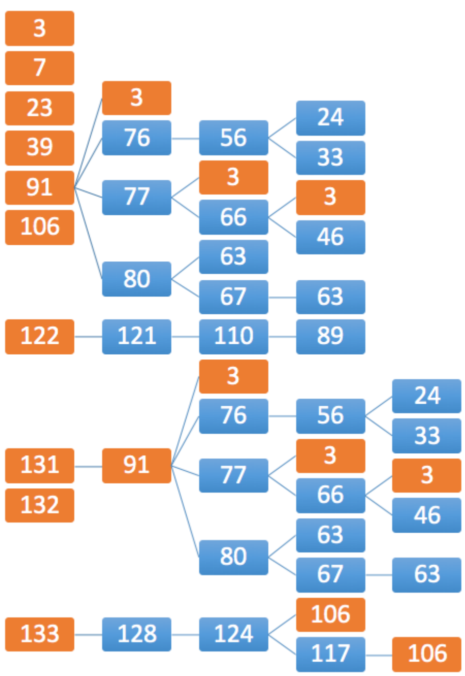

Figure 4: Evolution of solutions up to those in the tag cloud in the final generation. The leftmost 10 rectangles correspond to the final solutions and the other rectangles represent solutions that influenced the creation of the final 10 solutions. The lines between two rectangles indicate that the right solution influenced the left one.

As shown in Figure 4 and Table 5, we can trace the evolution of solutions if we have a tree structure representing the evolution and the list of solutions corresponding to the nodes and leaves. For example, solution number 3, "money", became one of the final solutions. We can observe that this solution existed from the early stage of the problem solving and influenced the creation of many other solutions. An analysis based on the evolution tree can reveal how problem solving relies on human creativity.

Visualizing the evolution of solutions should be useful not only after problem solving but also during problem solving. One answer to question 4 suggested that the system should visualize the evolution of solutions and show the evaluations of the quality of the entire problem solving process during problem solving in real time to better guide the problem solving process. We need to consider how to utilize the detailed information from the evolution of solutions during and after problem solving.

Table 5: List of solutions related to the final ones, namely target solutions to which crossover and mutation were applied. These solutions were originally stated in Japanese. Answers in brackets were originally proper nouns.

Number	Solution
3	money
7	meeting people who understand me
23	personal connections
24	music
33	(a type of food)
39	delicious food
46	(a person's name)
56	stimuli
63	(a game title)
66	free of pressure
67	(a game title)
76	sense of humor
77	taking a trip
80	(a game title)
89	credits for graduation
91	free time
106	(a companion animal)
110	human-based EC system
117	(a companion animal)
121	down quilts for the winter
122	kotatsu (a Japanese style of heater)
124	companion animals
128	(a character in a comic)
131	reading
132	(an anime title)
133	(a character in a comic)

5 CONCLUDING REMARKS

In the present study, we developed and evaluated an intuitive and traceable human-based EC system that is applicable to problems for which solutions are described in long sentences. In an experiment, participants could intuitively execute EC using the proposed system and the evolution of solutions could be traced in detail. To evaluate the system, we asked the human subjects to solve a problem using the developed system. The answers to a survey given to the human subjects after problem solving were subjected to statistical hypothesis testing. The results show that the system created better solutions than those created independently by the human subjects. We demonstrated that the system could trace the evolution of solutions by using the information obtained from solution creators during problem solving.

In future work, we will consider mechanisms for enhancing the motivation of participants to make them contribute more to problem solving. This is key for the wide use of human-based EC systems. We will consider showing the information obtained during problem solving, such as the evolution of solutions, to participants in real time. The participants could also be ranked based on their contribution to problem solving.

ACKNOWLEDGEMENT

We are most grateful to Masatomo Azumaya and Kosuke Fujimoto for their system developments and discussions. This work is supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (C)(18K11471).

REFERENCES

[1] M. Azumaya and K. Ohnishi. 2016. A Generation Gap Model for a Human-based Evolutionary Algorithm Using a Tag Cloud. In *Proceedings of Joint 8th International Conference on Soft Computing and Intelligent Systems and 17th International Symposium on Advanced Intelligent Systems (SCIS&ISIS2016)*. 880–885.

[2] C. D. Cheng and A. Kosorukoff. 2004. Interactive one-max problem allows to compare the performance of interactive and human-based genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*. 983–993.

[3] R. Hasebe, R. Kouda, K. Ohnishi, and M. Munetomo. 2014. Human-based Genetic Algorithm for Facilitating Practical Use of Data in the Internet. In *Proceedings of Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems (SCIS&ISIS2014)*. 1327–1332.

[4] R. Hasebe, K. Ohnishi, and M. Koeppen. 2013. Distributed Human-based Genetic Algorithm Utilizing A Mobile Ad Hoc Network. In *Proceedings of IEEE International Conference on Cybernetics (CYBCONF 2013)*. 174–179.

[5] A. Kosorukoff. 2001. Human based genetic algorithm. In *Proceedings of 2001 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2001)*. 3464–3469.

[6] A. Kosorukoff. 2002. Evolutionary computation as a form of organization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*. 983–993.

[7] X. Llora, K. Ohnishi, Y.-P. Chen, D. E. Goldberg, and Michel E. Welge. 2004. Enhanced innovation: A fusion of chance discovery and evolutionary computation to foster creative processes and decision making. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*. 1314–1315.

[8] J. Okano, K. Hamano, K. Ohnishi, and M. Koeppen. 2014. Particular Fine-Grained Parallel GA for Simulation Study of Distributed Human-Based GA. In *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2014)*. 3523–3528.

[9] J. Ren, J. V. Nickerson, W. Mason, Y. Sakamoto, and B. Graber. 2014. Increasing the crowd's capacity to create: how alternative generation affects the diversity, relevance and effectiveness of generated ads. *Decision Support Systems* 65 (2014), 28–39.

[10] H. Sayama and S. D. Dionne. 2015. Studying collective human decision making and creativity with evolutionary computation. *Artificial Life* 21, 3 (2015), 379–393.

[11] H. Wu, J. Corney, and M. Grant. 2015. An evaluation methodology for crowdsourced design. *Advanced Engineering Informatics* 29 (2015), 775–786.