Finding Good Affinity Patterns for Matchmaking Parties Assignment through Evolutionary Computation

Sho Kuroiwa[1,2], Keiichi Yasumoto[1], Yoshihiro Murata[3], and Minoru Ito[1]

[1] Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma, Nara, Japan
{sho-k,yasumoto,ito}@is.naist.jp

[2] Hopeful Monster Corporation, 8916-5, Takayama, Ikoma, Nara, Japan
sho-k@hopefulmonster.jp

[3] Hiroshima City University, 3-4-1 Ozuka-Higashi, Asa-Minami, Hiroshima, Japan
yosihi-m@hiroshima-cu.ac.jp

Abstract. There is a demand to maximize the number of successful couples in matchmaking parties called "Gokon" in Japanese. In this paper, we propose a method to find good affinity patterns between men and women from resulting Gokon matches by encoding their attribute information into solutions and using an evolutionary computation scheme. We also propose a system to assign the best members to Gokons based on the method. To derive good affinity patterns, a specified number of solutions as chromosomes of evolutionary computation (EC) are initially prepared in the system. By feeding back the results of Gokon to the solutions as fitness value of EC, semi-optimal solutions are derived. To realize the proposed system, we need simultaneous search of multiple different good affinity patterns and efficient evaluation of solutions through as small number of Gokons as possible with various attribute members. To meet these challenges, we devise new methods for efficient selection operation inspired by Multi-niches Crowding method and reuse of past Gokon results to evaluate new solutions. To evaluate the system, we used the *NMax* problem assuming that there would be N good affinity patterns between men and women as a benchmark test. Through computer simulations for $N = 12$, we confirmed that the proposed system achieves almost twice as many good matches as a conventional method with about half the evaluation times.

Keywords: Evolutionary Computation, Matchmaking Party, Multi-niches Crowding.

1 Introduction

Recently, the low birthrate has become a serious problem in Japan. One of the reasons for the problem is the lack of opportunities to find a marriage partner. For this reason, several local governments and enterprises have provided opportunities for unmarried people to meet potential marriage partners. In particular,

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part II, LNCS 7492, pp. 498–507, 2012.

matchmaking parties called “Gokon” are now attracting considerable attention in Japan. It is important for such a matchmaking party to assign participants so that the number of man and woman pairs likely to begin relationships is maximized. We regard this pair as a “good match”. However, assigning members to a Gokon so as to maximize the number of good matches is difficult since affinity between men and women is not yet well-understood (prediction problem), and determining the best Gokon members is also difficult (combinatorial optimization problem). In this paper, we propose a system to solve these two problems.

To resolve the prediction problem, Evolutionary Computation (EC) [1] is used. EC is a well-established method for solution search of the target system and has a wide range of applications [2]. The system has concatenations of man and woman attribute information (called the *attribute*, hereafter) as solutions of EC (chromosomes or individuals) to find semi-optimal solutions representing a good affinity by feeding back the number of good matches in a Gokon as fitness values. To resolve the combinatorial optimization problem, we set Gokons as many as possible and assign, to each Gokon, a specified number of men and women who have attribute similar to good affinity obtained as a solution. It is perhaps not the best way, but we focus mainly on resolving the prediction problem in this paper.

Since our target prediction problem is a multimodal problem, EC has to find many peaks (good affinity patterns) in the domain of affinity patterns. However, most of the existing application studies treat how to find the peak of a unimodal problem [2]. Our previous research also did not devise a method for solving multimodal problem [3]. So, the method inspired by Multi-niches Crowding [4] selection which has a reputation in the EC domain as a technique to calculate the multi-maximum of multimodal function is adopted in the proposed system. To find optimal solution of the target problem, an incredibly large number of Gokons are needed. Thus, the method using archival records to evaluate the solution instead of doing new Gokons is adopted.

We compared performance of the proposed system with the greedy approach using computer simulations. The volume of attribute and number of peaks in affinity domain is twice as large as in previous research [3], and the function of good match is defined taking into account the uncertainty in the real world at this time. Through computer simulations, we confirmed that the proposed system can generate almost twice good matches in half real Gokons compared to the greedy approach.

2 Gokon Problem

The Gokon problem is to divide a large population (system users) into small groups (Gokon) with the same number of men and women with good affinity where each user can be included in different Gokons until he/she makes a match.

Input: The input of the problem is the sets of male participants and female participants denoted by $B = \{b_1, b_2, \cdots\}$ and $G = \{g_1, g_2, \cdots\}$, respectively.

Each man $b_i \in B$ and each woman $g_j \in G$ have k attributes $(b_{i1}, b_{i2}, \cdots, b_{ik})$ and l attributes $(g_{j1}, g_{j2}, \cdots, g_{jl})$ respectively, where each attribute represents his/her feature and/or personality.

Output: The output of the problem is to make h groups (Gokons) that can be overlapped, where each Gokon has L men and L women from B and G.

Number of Gokons is h, and the i-th Gokon is represented by $[B_i, G_i]$. The output is Gokon assignment $[B_1, G_2], \cdots, [B_h, G_h]$ to optimize objective function and unknown evaluation function F.

Objective Function: Evaluation function F, which returns the number of good matches in the Gokon $[B_i, G_i]$, is defined. Objective function is to maximize the sum of F for all h Gokons, and is given by

$$\textbf{maximize} \sum_{i \in \{1,\ldots,h\}} F(B_i, G_i) \qquad (1)$$

3 Proposed System

We propose a system to evolve the evaluation function of the Gokon problem using EC and compute Gokon participant lists as shown in Fig. 1. The system is composed of **solutions** representing good affinity between men and women and the following three operators. The **EC operator** executes EC calculation (Selection, Crossover and Mutation) to the solutions and generates **candidate solutions** (indicated in Fig. 1 as Step.1). The **Assignment Operator** generates a participant list corresponding to each candidate solution (indicated in Fig. 1 as Step.2). The **Evaluation Operator** gives fitness value to each candidate solution (indicated in Fig. 1 as Step.3). Finally, we update the solutions using candidate solutions. Running through these steps, solutions will improve gradually toward optimal solutions. Here, we define initial solutions as first generation and the solutions updated $t-1$ times as t-th generation. At the evaluation of candidate solution, we want to reduce the number of Gokon times because the optimal participant lists cannot be obtained until the system finds good affinity. Until this point, the system inflicts a time and money loss on participants to make good matches. So, it is desirable to be able to evolve candidate solutions without doing actual Gokons. Then, we devise a method using archival records of attributes of good matches indicated in Fig. 1 as **Stock**.

3.1 Solution

Each solution of EC is coded as a concatenation of attribute of a man and a woman who are likely to make a good match. The first half of each solution shows the attribute of a man, and the remaining half shows that of a woman as in Fig. 2.

Input of the system, man and woman attribute correspond to each half of the solution. Those attributes are obtained by the system from a user questionnaire and personality test, the Temperament and Character Inventory (TCI) [5].

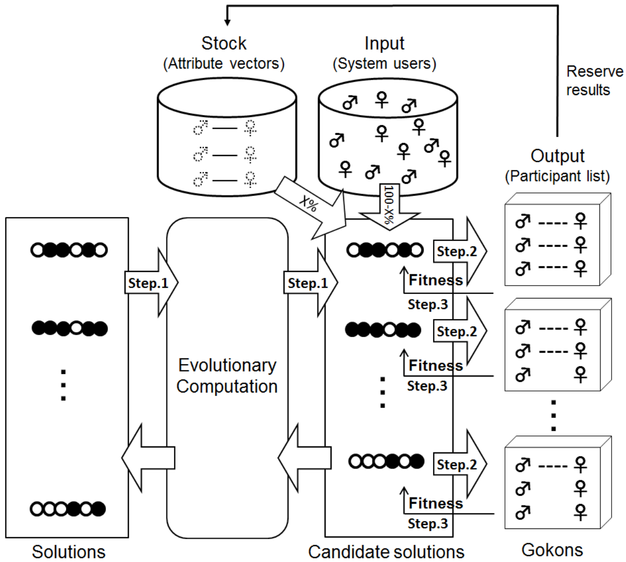

Fig. 1. Outline

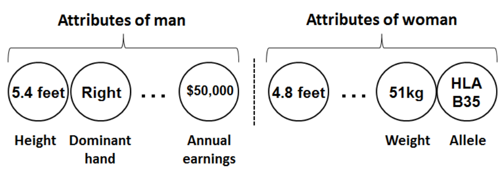

Fig. 2. Example of solution representation

However, what information is important for determining an affinity is not known well academically, and matchmaker companies use slightly different information on their own. For example, TCI has many question items but the result of the test is categorized into 12 patterns (it takes only 4 bits). To determine the element factor of affinity, we estimate that the necessary information size to encode the solution is down to 20 bits each for man and woman (totally 40 bits).

3.2 EC Operator

The EC Operator performs three operations in solutions: selection, crossover and mutation. We adopted one-point crossover and mutation that are commonly used as the EC operations.

However, the Gokon problem is a multimodal function with multiple good affinities, and solutions need to have diversity to search for multi-optimal so-

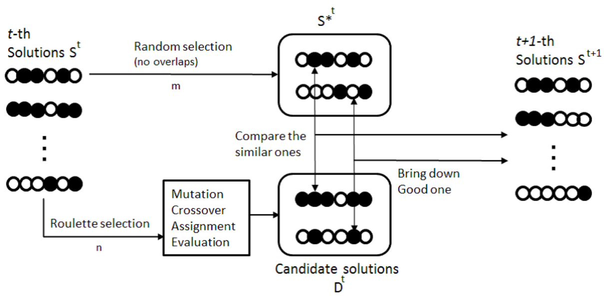

Fig. 3. Scheme of Multi-niches Crowding Factor

lutions. Therefore, we adopted a selection method considering Multi-Niches Crowding [4](hereafter, we call this the MNC method), which is one of the representative methods for keeping diversity in solutions. The MNC method prevents the increase of similar solutions in subsequent generations.

In the proposed method, first we select randomly m ($m \leq |S^t|$) solutions as $S*^t$ without overlapping from S^t as shown in Fig. 3. Second, we select n ($n \leq |S^t|$) solutions by roulette selection from t-th solutions S^t. The we apply EC operations, crossover and mutation, assignment and evaluation to those, and get D^t ($|D^t| = n$). Finally, we find the most similar $s* \in S*^t$ for each $d \in D$ and bring down d or $s*$, which has the higher fitness value, into next generation S^{t+1}.

3.3 Assignment Operator

This operator assigns members into the Gokon corresponding to each candidate solution (generate participant list of Gokon). If a man and woman have the same attributes as a candidate solution and make a good match, it is natural and preferable to use this result in calculating a fitness value of this candidate solution. However, there are very few men and women who have exactly the same attributes as the candidate solution. So, this operator assigns a specified number of men and women in the order of **similarities**. After this operation, participants are ready to start the Gokon.

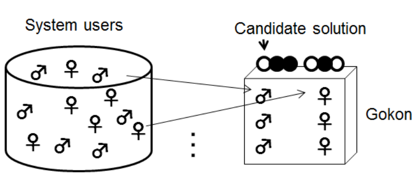

Fig. 4. Operation of Assignment

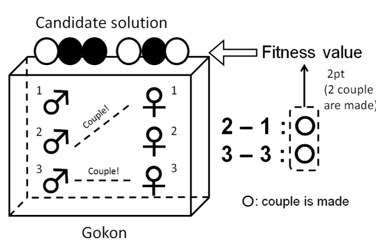

Fig. 5. Operation of Evaluation

3.4 Evaluation Operator

The evaluation operator gives results of the Gokon as a fitness value of the corresponding candidate solution. These results are obtained by the participants joining each Gokon through a questionnaire. After a Gokon, each participant answers the questionnaire about the participants of the opposite sex. We use the number of good matches in the Gokon as fitness value of the solution, with adjustment considering similarities (described in Sect. 3.3) between their attributes and candidate solution. That means if a candidate solution is not similar to the attributes of man and woman who make a good match, the match should not contribute to the evaluation of the solution.

Then, we prepare the new index of the similarities called **matchrate**. Candidate solution is denoted by vector $\mathbf{s} = (s_1, \cdots, s_{k+l})$. We define a pair of man and woman as (b, g), where attributes of b are $(b_1, b_2, \cdots, b_k)$ and g are $(g_1, g_2, \cdots, g_l)$, a vector concatenating those attributes is $\mathbf{x} = (b_1, b_2, \cdots, b_k, g_1, g_2, \cdots, g_l)$ denoted by $(x_1, x_2, \cdots, x_{k+l})$. Hereafter, we call this $\mathbf{x}$ the attribute vector. At this time, matchrate C is given by

$$C(\mathbf{s}, \mathbf{x}) = \frac{\sum_{i=1}^{k+l} match(s_i, x_i)}{k+l} \tag{2}$$

Here, *match* is defined as follows.

$$match(s_i, x_i) = \begin{cases} 1 & \text{when } (s_i = x_i) \\ 0 & (otherwise) \end{cases} \tag{3}$$

Thus, the sum of the matchrate value in the Gokon is given as the fitness value of the candidate solution as shown in Fig. 6. The fitness value is going to be 0 when there are no good matches.

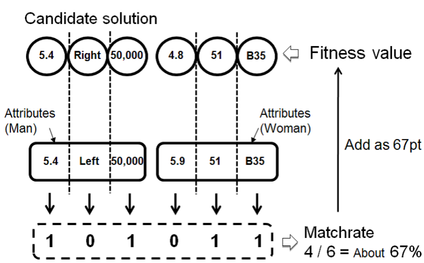

Fig. 6. Calculation of matchrate

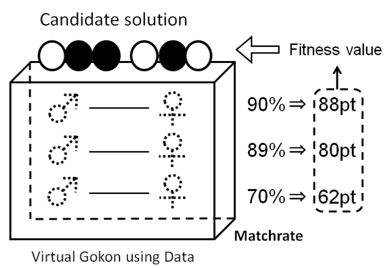

Fig. 7. Evaluation using stock

Furthermore, using this matchrate, we can give fitness value using archival records of good matches without doing a real Gokon. As shown in Fig. 1, we reserve attribute vectors of good matches, called stock, in the early phase. When evaluating the candidate solution, we apply the assignment operator to the stock and real users in the ratio of $X\%$ to $100 - X\%$. We regard data from Stock as attribute vectors of real users for evaluating the candidate solutions as shown in Fig. 7.

4 Experiments

We conducted a computer simulation to evaluate how efficiently the proposed system can solve the Gokon problem. For evaluation, we defined a benchmark test called the *NMax* problem [3].

In the experiment, first we want to confirm if the MNC method works well. Thus, we compared the solutions obtained with and without MNC. We define **the optimum achievement rate** as a metric to evaluate the solutions for this purpose.

We also measured the total number of good matches for all Gokons using solutions at $1,000$-th generation. We compared the results between the proposed method and the greedy method, which is the conventional way to greedily search for good affinity patterns (i.e., peaks).

4.1 The Benchmark Test (*NMax* Problem)

The *NMax* Problem is the problem where N arbitrary bit strings represent the solutions [3].

Input: Input of *NMax* Problem is attribute vector $\mathbf{x}$ defined in Sect. 3.4 and N peak vector $P = (\mathbf{p}^1, \cdots, \mathbf{p}^N)$ where $\mathbf{p}^i = (p_1^i, \cdots, p_{k+l}^i)$. In the experiment, $b = (b_1, b_2, \cdots, b_k)$ and $g = (g_1, g_2, \cdots, g_l)$ are given randomely like $(1\ 0 \cdots 1)$.

Output: Output of *NMax* is given as follows.

$$f_{NMax}(\mathbf{x}) = max_{1 \le j \le N}(match(\mathbf{x}, \mathbf{p}^j)) \tag{4}$$

Here, j is the index of N which indicates the peak of *NMax* and $match(\mathbf{x}, \mathbf{p})$ is defined as Eq.(3). The output of the *NMax* Problem is the degree of how close the input vector $\mathbf{x}$ is to one of the peaks of the *NMax* problem. When $N = 1$, $\mathbf{p}^1 = \{1, 1, \cdots, 1\}$, the *NMax* Problem is identical to the *OneMax* Problem. We make a strong assumption that the *NMax* Problem can represent the existence of N good affinity patterns (attribute vectors) in real-world.

In this experiment, we improve the *NMax* Problem to be more realistic. In real-world Gokon, human decision is often affected by the situation or the state of mind. So, we add the randomness into the *NMax* function. The function G, output of *NMax* Problem, returns the probability of making a good match by using f_{NMax} as follows.

$$G = \begin{cases} 1 - \frac{1 - f_{NMax}}{1 - \beta} & (\beta \le f_{NMax}) \\ 0 & (otherwise) \end{cases} \tag{5}$$

Here, β is the threshold to make a good match. When output of f_{NMax} is smaller than β, probability of a good match G equals 0, while for f_{NMax} more than β, P increases linearly.

4.2 Optimum Achievement Rate

To evaluate solution $\mathbf{s}$, using optimal solutions $\mathbf{p}^i$ indicated in Fig. 9 and *match* function defined as Eq.(3), the optimum achievement rate R is given by

$$R = max_{1 \leq i \leq n}(match(\mathbf{s}, \mathbf{p}^i)) \tag{6}$$

This is the degree of how close the solution is to one of the optimal solutions of the *NMax* problem. So, the average value of the solutions described as bellow is key to understand how close the solutions are to good affinities.

$$\overline{R} = \frac{\sum_{i=0}^{|S^t|} R_i}{|S^t|} \tag{7}$$

4.3 Comparative Method

In the greedy method, the candidate solution is updated to its most similar attribute vector of man and woman who make a good match as shown in Fig. 8 and bring down solutions which has the higher fitness value into next generation, instead of EC operation. This method intends to improve the solutions gradually as generations progress.

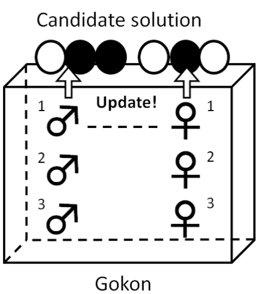

Fig. 8. Greedy Method

Peak.1	0000000000000000000000000000000000000000
Peak.2	0000000000000000000011111111111111111111
Peak.3	0000000000111111111100000000001111111111
Peak.4	1111100000111110000011111000001111100000
Peak.5	1111100000111110000000000111110000011111
Peak.6	1010101010101010101010101010101010101010
Peak.7	0101010101010101010101010101010101010101
Peak.8	0000011111000001111111111000001111100000
Peak.9	0000011111000001111100000111110000011111
Peak.10	1111111111000000000011111111110000000000
Peak.11	1111111111111111111100000000000000000000
Peak.12	1111111111111111111111111111111111111111

Fig. 9. *NMax* Problem ($N = 12$)

4.4 Experimental Setup

Input Data: The number of participants is $6,000$ ($3,000$ men and $3,000$ women), length of attribute of participant ($|b_{ik}|$, $|g_{jl}|$) is 20 bits (so that attribute vector and solution length are 40 bits each), the number of attribute vectors in stock is 500, the threshold β to make a good match is 0.5 and Gokon size is 30 ($|B_i| = 15, |G_i| = 15$). Each user can only participate in Gokon five times. The system replaces the user who made good match or participated Gokon five times with a new user in every generation.

The EC Parameters: The number of solutions is 30, the selection rate is 0.5, the crossover rate is 0.95, the mutation rate is 0.2. The solutions are initially generated by uniform random number.

In this experiment, if there exist no man and woman who have attribute similar to the peak of *NMax* indicated in Fig. 9, the number of good matches will dramatically decrease. This prohibits the progress in evolving solutions. Thus, we prepare new attribute randomly which has 1 to 100% similarity to each peak.

In general, it is important to consider what attributes are needed and how the attributes are coded as we described in Sect. 3.1. However, in this experiment, we focus on investigating whether our proposed algorithm can effectively find optimal solutions (i.e., N good affinities).

4.5 Results

We show the optimum achievement rate for $12Max$ Problem at 1,000-th generation in the cases using MNC and not using MNC, respectively in Figs. 10 and 11. The optimum achievement rate $\overline{R}$ is shown as an average value of solutions.

```
0000000100110000001001000100010000000010001 Peak:1  77%
0000011100100010100000000110010010000100 Peak:1  72%
0010010110001000000000000001000110000000 Peak:1  80%
0000001000000100000000000101001000001100 Peak:1  82%
0011000100010011100011110111111111110111 Peak:2  77%
0001000000000000001001111111111111011111 Peak:2  92%
0000100001000000000000110111010101011111 Peak:2  80%
0010000000100000000000110101111111111111 Peak:2  90%
0000011100000000000011011101101110111111 Peak:2  80%
0000100010111111101101000010000111111100 Peak:3  80%
0000000000111100111100100000001011010111 Peak:3  85%
1101100001101110000010110100111111100000 Peak:4  80%
0111100000111101010011111011001101100000 Peak:4  82%
0111100010111111010011110000001111010000 Peak:4  82%
1111101010111110110000000110010001011101 Peak:5  80%
1011100100001100000000000011101110001011 Peak:5  77%
0111100001111010000000000011001011011111 Peak:5  80%
1010101110101010100010101110101010100000 Peak:6  87%
0101011111000101010101011111001001010101 Peak:7  80%
0101110101010000100101000111010001010101 Peak:7  80%
1000011110100001010010111100000111100000 Peak:8  77%
0000011111010001110110010000011011100000 Peak:8  82%
0000111101011001111111111100000111100000 Peak:8  85%
0000000111000100111110100110000100011111 Peak:9  75%
1100000010000011011100100111110000011111 Peak:9  77%
0000011011101001111100010110100000011110 Peak:9  82%
0000111110001000111100100111110000010111 Peak:9  85%
1100111101010111111100000001000000000001 Peak:11 82%
1110111111101100110100100010010001000000 Peak:11 77%
1111101011110110101111110111011101111111 Peak:12 80%
```

Fig. 10. Using MNC ($\overline{R} = 81\%$)

```
1110001010000010001100100011000011000111 Peak.1  60%
0000101000010100011110110101001011000111 Peak.2  60%
0011001000010010111111101101011110010111 Peak.2  62%
1100000010100111001000100111000111100111 Peak.3  62%
1101001000100110111100110101000111100010 Peak.3  60%
1110001010101111101100010000000111100111 Peak.3  75%
1110100000110110001100110011100111100100 Peak.4  70%
1111101000100111000100110001000110101110 Peak.4  65%
1001001100110010100001111111001100100100 Peak.4  62%
1111000110000101110011101000101110001101 Peak.4  57%
1010110001110110011101001010111000110110 Peak.5  60%
1011001100110111010111110101011000011111 Peak.5  62%
1010001110101110001100110111100110100100 Peak.6  62%
1110011010001110001111110011100100100110 Peak.6  60%
1010001010111111001100100011100101100010 Peak.6  65%
1111110100110101011110110101001101011010 Peak.7  62%
1100011110000111011011111010110111001110 Peak.8  62%
1010011101100001111011000010101010001101 Peak.9  65%
1100001111101001011001011011011010101111 Peak.9  60%
0000111011101000000101001101110100001110 Peak.9  65%
1011100111000000111101010110110010101011 Peak.9  65%
1000111011100101110000101111010011010111 Peak.9  67%
0001001111010110101111010011100110111101 Peak.9  60%
1110110001110000001110101101100001101001 Peak.10  60%
0110110101000010000010011110010100111001 Peak.10  65%
1101101111000011011010101101000110011100 Peak.10  60%
0111011010010111111001101011010010101000 Peak.11  60%
0101101011111011111111111110000101001010 Peak.12  65%
0111101111110000111111101110001100001110 Peak.12  62%
1111011010000111001110110010111111010110 Peak.12  62%
```

Fig. 11. Not using MNC ($\overline{R} = 63\%$)

As a result, the method using MNC can obtain about 18% higher value than the method not using MNC.

As shown in Table 1, optimum achievement rate is obtained after 1,000 EC generations, and number of couples (good matches) is calculated by applying an assignment operator to the solutions at the 1,000-th generation. Here, $\overline{R}$ is shown as an average value of 30 trials, and plus-minus means the standard deviation which started from different initial solutions generated randomly. Proposed (Stock use: 50%) in Table 1 is the case using stock. In this case, the total times of real Gokon is 5,250, almost half the times as without stock (stock use: 0%).

Table 1. Comparison of proposed and greedy method

	$\overline{R} \pm S.D.$	Number of couples	Number of real Gokon
Optimal*	100 ± 0%	6302 ± 24 pairs	0 Times
Proposed (Stock use: 0%)	82 ±1 %	5003 ± 102 pairs	10,000 Times
Proposed (Stock use: 50%)	80 ± 1 %	4698 ± 132 pairs	5,250 Times
Greedy	64 ± 7 %	2530 ± 334 pairs	10,000 Times

* At the case when giving optimal solutions (Fig. 9) into solutions.

5 Conclusions and Future Work

In this paper, we defined the Gokon problem, which assigns men and women who are likely to make good matches in the same Gokon. We also proposed the Evolutionary System to solve the problem and evaluated the system using computer simulation. For the 12*Max* problem, which is the problem of finding 12 unknown peaks (i.e., good affinity patterns), we confirmed that the proposed system achieves 82% similarities to optimal solutions and almost twice as many good matches as the conventional method with about half the evaluation times.

As part of future research, we plan to fill the gap for applying the proposed system to actual Gokons. To obtain the results of this experiment, we needed Gokon data for 5 to 10 thousand times. Major companies in this domain in Japan perform Gokons about 2 to 15 thousand times a year. Thus, if we can use such data, the system will be available within a more practicable timeframe.

References

1. Foster, J.A.: Evolutionary Computation. Nature Rev. Genet. 2, 428–436 (2001)
2. Cagnoni, S., Poli, R., Smith, G.D., Corne, D., Oates, M., Hart, E., Lanzi, P.L., Willem, E.J., Li, Y., Paechter, B., Fogarty, T.C. (eds.): Real-World Applications of Evolutionary Computing. LNCS, vol. 1803. Springer, Heidelberg (2000)
3. Kuroiwa, S., Murata, Y., Kitani, T., Yasumoto, K., Ito, M.: A Method for Assigning Men and Women with Good Affinity to Matchmaking Parties through Interactive Evolutionary Computation. In: Proc. of Simulated Evolution and Learning, pp. 645–655 (2008)
4. Cedeno, W., Vemuri, V., Slezak, T.: Multi-Niche Crowding in Genetic Algorithms and its Application to the Assembly of DNA Restriction-Fragments. Evolutionary Computation 2(4), 321–345 (1995)
5. Cloninger, C.R., Svrakic, D.M., Przybeck, T.R.: A Psychobiological Model of Temperament and Character. Archives of General Psychiatry 50, 975–990 (1993)