Global Optimization Methods for Assigning Collaborators to Multiple Problems Using Genetic Algorithm

Gunwoo Kim
Hanbat National University
gkim@hanbat.ac.kr

Keunho Choi*
Korea University
keunho@korea.ac.kr

Yongmoo Suh
Korea University
ymsuh@korea.ac.kr

Abstract

As firms continuously come across new and complex problems in the fast-changing business environment, they have to find available collaborator with proper expertise that is required by a problem and assign them to the problem which requires their expertise, so that the best results can be reached from the firm’s holistic point of view. Since this kind of an optimization problem is taking place continuously in a firm as the business environment changes, Genetic Algorithm (GA) which has shown outstanding performance in obtaining a sub-optimal solution relatively fast seems to be the right choice for such an optimization problem, instead of the other approaches such as goal-programming, multi-attribute decision making, and branch and bound. Therefore, we propose a GA-based approach to solving the above problem of assigning collaborators to multiple complex problems as well as illustrate our proposed method with an example.

1. Introduction

In these days, business environment is rapidly changing. As a result, firms continuously come across new problems whose complexity is beyond their capability. Therefore, they need to seek collaborators with expertise from the outside, if they cannot find them internally, and similar situation also may take place within a firm. That is, there may be several departments each of which has some problems, solutions of which cannot be obtained within the department, but there are many available employees in other departments who can make collective contribution in solving the problems. In this situation, the role of human resource department is to find available employees with proper expertise that is required by a problem and assign them to the problem which requires their expertise, so that the best results can be reached from the firm’s holistic point of view.

This kind of problem is an optimization problem which has been dealt with by diverse approaches such as goal-programming [7], fuzzy multi-objective programming [10], multi-attribute decision making [2, 12], particle swarm optimization [15], branch-and-bound algorithm [9] and genetic algorithm [4]. Especially, the genetic algorithm (GA) has been widely used as a new approach to optimization problem in many areas because of its powerful capability in obtaining a sub-optimal solution in a reasonably short time, and has achieved satisfactory results [4-5, 8, 13]. Thus, GA is appropriate in case of finding collaborators with proper expertise for multiple complex problems that take place in the fast-changing business environment, which is dealt with in this paper.

The objective of the paper is to propose a GA-based solution to the above problem of assigning employees with proper expertise (we call them collaborators, as they have to collaborate to solve a complex problem) to multiple complex problems. This optimization problem has several requirements, including these. Firstly, each problem requires an expertise in one or more subjects (problem requirement). Secondly, each employee should have an expertise which satisfies all or at least parts of the requirements of some problems (employee requirement). Both the problem requirement and employee requirement can be described in terms of subjects. Lastly, the solution should be globally optimal in that it leads to the best situation from the firm’s holistic point of view as well as it satisfies the requirements of all problems (firm requirement).

The rest of the paper is organized as follows. Section 2 reviews previous studies regarding the optimization of partner selection. Section 3 describes the research framework for realizing our approach and provides a detailed description of each step of the framework. Section 4 illustrates our approach with an example. The last section contains concluding remarks.

* Corresponding Author,
Tel: +82 2 3290 1945/ Fax: +82 2 922 7220
E-mail address: keunho@korea.ac.kr
Postal code: 136-701
including a summary, implications, and limitations of the paper.

2. Literature review

In relation to our research problem of selecting collaborators for each of the problems which each department has, previous researches on ‘partner selection problem’ have been examined. Most previous researches on partner selection have been conducted in virtual enterprise or international joint venture environment [3, 7-9, 11-12, 14-15], for supply chain formation [1-2, 10], or for team formation [4-6].

Hajidimitriou and Georgiou [7] proposed a goal programming-based quantitative model for evaluating candidate partners and selecting only one optimal partner in international joint ventures. In their study, profit, financial ratio, composite index, and other twelve goals were considered, and the mathematical formulations of the goals including quantitative and qualitative criteria were presented.

To obtain the solution for sub-contractor selection in an agile manufacturing environment, Ip et al. [9] first reduced the solution space by defining and removing inefficient sub-contractors and then embedded a branch and bound algorithm into project scheduling by defining the ideal candidate. But the algorithm has a difficulty in solving large scale problems because of its processing complexity [9]. One difference between their and our research is that they considered several sub-projects of a project while we considered multiple independent problems. Another difference between the two is that a candidate can be selected for more than one sub-project without limit, while a collaborator should be selected for maximum X number of problems in our approach.

Ye and Li [12] proposed two group multiple attribute decision making (MADM) methods to select suitable partners of a virtual enterprise. They used interval values to represent the incomplete and uncertain information about candidate partners. In their study, five attributes such as cost, time, trust, risk, and quality were used as criteria for selecting partners. However, they did not take into account characteristics of problems that the virtual enterprise deals with in partner selection process.

Zhao et al. [15] presented a model for partner selection with precedence requirement and due-date constraints in a virtual enterprise. They assumed that an enterprise wins a large project consisting of a set of subprojects, and the enterprise decides to select partners for each subproject because of finite resources and capabilities. As a means to select partners under the assumption, they used particle swarm optimization (PSO) and adapted the PSO for further improved performance using three mechanisms – initialization expansion, variance, and local search.

Chang et al. [2] used a fuzzy multi-attribute decision making (FMADM) method which applies fuzzy concept to both the ordinal and the cardinal information to develop a supplier selection procedure. By considering different phases in product life cycle, they attempted to satisfy current product competition strategies and to improve effectiveness and efficiency of the entire supply chain.

Chamodrakas et al. [1] suggested an approach to selecting effective suppliers in an electronic marketplace. They proposed pre-qualification stage to reduce the search space for supplier selection and focus only on the most suitable suppliers. In that stage, each supplier is assessed and rated for each quantitative and qualitative criterion from a five-point scale. Then, suppliers who have higher value than the threshold of each criterion are considered in the final evaluation stage, in which they introduced a modified Fuzzy Preference Programming (MFPP) method which combines FPP with the rating scale Analytic Hierarchy Process (AHP). The MFPP enabled them to overcome the explosion of the number of pair-wise comparisons that could happen when there are many suppliers and/or criteria, and to solve the problem of inconsistency caused by being unsure of decision makers’ preferences.

Wu et al. [10] proposed a fuzzy multi-objective programming model to select supply chain partners taking into account both quantitative and qualitative risk factors. Quantitative risk factors include factors such as cost, quality, and logistics, and qualitative risk factors such as economic environment and supplier’s service rating evaluated by customers. In addition, they dealt with five objectives such as minimization of the total purchase cost, the number of rejected items, the number of late deliveries, risk factors of economic environment, and risk factors of supplier service rating.

Fan et al. [4] and Feng et al. [5] used GA to select members for R&D teams and cross-functional teams, respectively, with individual and collaborative information. Fan et al. [4] considered both individual information such as research experience, communication activity and academic reputation, and collaborative information such as the number of papers published with others and the number of projects accomplished with others. As a result, they found 16 pareto-optimal solutions. Similarly, Feng et al. [5] considered work experience, work capability, and specialized knowledge as measures for individual information, and communication, cooperation, and exterior organizational collaborative situation as measures for collaborative information. By using both
kinds of information, they were able to find 22 pareto-optimal solutions.

Ip et al. [8] proposed a rule-based genetic algorithm (R-GA) for identifying the optimal combination of partner enterprises for all sub-projects attempting to maximize the success of the whole project. They considered the success probability and processing time of candidate partners and reduced the solution space by removing inefficient candidate partners under the constraint that only one partner should be selected for each sub-project.

As is mentioned earlier, our study aims to solve simultaneously multiple problems each with different requirements by assigning collaborators with their own expertise to each of the problems. We propose a GA-based global optimization method for assigning collaborators to multiple problems, taking into account the requirements of individual problems and expertise of individual collaborators.

3. Model development

This section presents the description of a research problem to which our proposed method is applied, research framework and the detailed description of each step of the framework.

3.1. Research problem

Our research is motivated from the situation described in the introduction section. That is, human resources department of a firm is trying to assign available employees (collaborators) to new complex problems (of other departments) which require their expertise (e.g., skills, experience and knowledge), so that the firm can enjoy the best performance as a whole.

Our research problem has the following requirements, the first three of which are mentioned in the introduction section:

1) Problem requirement: each problem requires an expertise in one or more subjects.
2) Collaborator requirement: each collaborator should have an expertise which satisfies all or at least a part of the requirements of some problem.
3) Firm requirement: the solution should be globally optimal in that it leads to the best situation from the firm’s holistic point of view.
4) Subject requirement: each subject of a problem should be dealt with by at least one collaborator.
5) Department requirement: there are a maximum number of collaborators who could be assigned to each problem, as is requested by the owner department of the problem.
6) Working condition requirement: each collaborator should be assigned to maximum X number of problems.

Although it is possible to find the best collaborators for a complex problem or to find the best complex problem for a collaborator, we want to find best combinations of problems and collaborators, so that a global sub-optimal solution can be obtained from the firm’s point of view.

3.2. Research framework

Here, we describe the overall process of assigning collaborators to each problem assuming the situation mentioned in Section 3.1, as is shown in Figure 1.

Firstly, each department analyzes its problems to find appropriate subjects associated with each problem, prior to asking human resources department to find employees having expertise in the areas of the subjects. As a result, a profile for each problem is constructed with subjects associated with the problem.

Secondly, having examined the problem profiles, the human resources department finds available employees who have expertise in the areas of the subjects to build a collaborator profile for each employee. Note that both problem profile and collaborator profile consist of subjects. In our study, if a collaborator has expertise on a subject or if a problem has an associated subject, it is represented as a real number in its position in the collaborator profile or in the problem profile, respectively. Thirdly, correlations between every two subjects are calculated, so that we can replace a collaborator having an expertise on a specific subject with a collaborator having an expertise on another subject which is highly correlated with the specific subject, when we cannot find the former collaborator.

Finally, GA process is conducted using the problem and the collaborator profiles. As is usual in GA process, a population of chromosomes is generated, fitness value is computed for each chromosome and then if there is no satisfactory chromosome in the current generation, new chromosomes are generated repeatedly by crossover or mutation until a satisfactory one is found.

3.3. Profile building

After each department finds appropriate subjects for each of its problems, a problem profile is constructed for each problem with the subjects. The problem profile for a problem $p$, $P_p$, is represented as follows.
\[ P_p = \{w_{ps} | s = 1, ..., N\}, \quad \text{s.t.} \sum_{s=1}^{N} w_{ps} = 1 \]

, where \( w_{ps} \) denotes the weight on subject \( s \) in problem \( p \), and \( N \) denotes the total number of subjects.

Similarly, after the human resources department interviews with collaborators, a collaborator profile is constructed for each collaborator based on the subjects included in problem profiles. The collaborator profile of a collaborator \( c \), \( P_c \), is represented as follows.

\[ P_c = \{w_{cs} | s = 1, ..., N\}, \quad \text{s.t.} \sum_{s=1}^{N} w_{cs} = 1 \]

, where \( w_{cs} \) represents the degree of expertise of the collaborator \( c \) on subject \( s \).

Figure 1. Overall framework of the proposed method

### 3.4. Correlations between subjects

Suppose that there are no collaborators who have an expertise on a specific subject of a complex problem. In this situation, we cannot assign suitable collaborators to the problem, thereby making the problem remain unsolved. Alternatively, it can be solved by identifying subjects that are similar to the specific subject, finding collaborators who have expertise on the similar subjects, and assigning them as a proxy of those who have expertise on the specific subject.

To do this, we calculated the correlations between every two subjects by adapting cosine similarity measure, as shown in the left part of Eq. (1), where \( A \) denotes the number of problems and \( C \) the number of available collaborators. In addition, we have taken into account that the correlation between two subjects may be more reliable if both subjects appear more frequently in the problem and collaborator profiles. Therefore, we defined a new term, degree of match, as \( d_{sij} / D_{si} \), where \( d_{sij} \) denotes the number of problems and collaborators which include both subjects \( s_i \) and \( s_j \), and \( D_{si} \) represents the number of problems and collaborators which include subject \( s_i \). The adapted cosine similarity is multiplied by the degree of match in order to calculate the correlation between subjects \( s_i \) and \( s_j \), \( C(s_i, s_j) \).

\[
C(s_i, s_j) = \frac{\left( \sum_{p=1}^{A}(w_{ps_i})(w_{ps_j}) + \sum_{c=1}^{C}(w_{cs_i})(w_{cs_j}) \right)}{\sqrt{\sum_{p=1}^{A}(w_{ps_i})^2 + \sum_{c=1}^{C}(w_{cs_i})^2} \sqrt{\sum_{p=1}^{A}(w_{ps_j})^2 + \sum_{c=1}^{C}(w_{cs_j})^2}} \times \frac{d_{sij}}{D_{si}} \quad (1)
\]
3.5. GA for global optimization

In this section, we explain the details of the proposed GA-based approach to assigning collaborators to each problem, including chromosome representation, crossover and mutation, and fitness function.

3.5.1. Chromosome representation: In an implementation of the genetic algorithm, it is important how to represent the chromosome, which encodes a potential solution for an optimization problem and consists of multiple genes. Since each department requests a maximum number of collaborators who could be assigned to each of its problems, we can compute the maximum number of collaborators who should be assigned to all the problems. If the number is \( M \), each chromosome consists of \( M \) genes, each of which is represented as an integer, representing a collaborator. Therefore, the number should be less than or equal to the number of available collaborators.

Suppose that there are two departments each of which has problem 1 and 3, and problem 2 and 4, respectively. Also, suppose that each of the problems requires maximum 1, 3, 2, and 4 collaborators, respectively, and that there are 20 collaborators. Then, each chromosome consists of 10 genes and each gene is represented as a non-negative integer less than 21. Figure 2 shows one possible chromosome, which indicates that collaborator 6 is assigned to problem 1, collaborators 2 and 15 to problem 2, collaborators 3, 12 and 8 to problem 3, and collaborators 4, 9, 20 and 7 to problem 4.

**Figure 2. An example representation of possible chromosome**

3.5.2. Crossover and mutation: In the first step of GA process, an initial population of a fixed number of chromosomes is randomly constructed. Then, two parent chromosomes in the population are selected as operands of crossover operator to generate new child chromosomes for the next population. The operator first divides the parent chromosomes at a few points of the chromosome and then exchanges the segments in the same position of the two chromosomes to generate child chromosomes. To lower the probability of being a local optimization, mutation operator randomly changes the values of genes and produces a new chromosome. New populations of chromosomes are generated repeatedly until a stopping condition is met. In fact, it cannot find an optimal solution of a problem but a sub-optimal solution.

In our implementation, two-point crossover is employed and a mutation operator is applied to a chromosome after every 10 executions of crossover operator. It should be checked whether a new chromosome generated by either crossover or mutation operator violates the working condition requirement or not, whenever a new chromosome is generated.

3.5.3. Fitness function: Every time when a new chromosome is generated either by crossover or mutation operator, we must evaluate the chromosome in order to answer the question, “How good is the chromosome as an answer to our problem?”. A fitness function is defined as a criterion of the goodness of a chromosome. In our implementation, the goodness of a chromosome is represented as the extent to which the problem requirement and firm requirement are met by the chromosome. Then, the fitness value of each chromosome can be calculated as follows.

\[
\sum_{p=1}^{A} \frac{1}{N_p} \sum_{i=1}^{N} w_{ps_i} - \frac{\sum_{j=1}^{N} \sum_{i=1}^{h} w_{ps_i} \times C(s_i, s_j) \times w_{sj}^p}{\sum_{j=1}^{h} \sum_{i=1}^{h} w_{ps_i} \times C(s_i, s_j) \times w_{sj}^p}
\]

s.t. \( \sum_{j=1}^{N} w_{ps_i} \times C(s_i, s_j) \times w_{sj}^p > 0, \forall s_i \in S_p \) (2)

, where \( w_{sj}^p \) denotes the sum of \( w_{cs_j} \) for each collaborator who is assigned to problem \( p \), \( S_p \) a set of subjects associated with problem \( p \), and \( N_p \) the number of subjects in \( S_p \). The constraint in Eq. (2) comes from the subject requirement. As is formulated in Eq. (2), sum of mean absolute errors for each problem is used to evaluate the extent of match between the profile of a problem and overall profile of collaborators assigned to the problem, for all problems. Thus, a chromosome whose fitness value is smaller represents a better solution.

4. Illustrative example

This section illustrates with an example how we implemented GA to solve the problem of recommending collaborators for multiple problems. We started with the assumptions that a department \( d_1 \) has one problem (i.e., \( p_1 \)) and another department \( d_2 \) has two problems (i.e., \( p_2 \) and \( p_3 \)), that ten collaborators are available and that \( d_1 \) requested...
maximum two collaborators for $p_1$ and $d_2$ requested maximum three collaborators for $p_2$ and $p_3$ (i.e., department requirement). We also assumed that each collaborator can be assigned to maximum two problems (i.e., working condition requirement). Problem profile given in the upper part of Table 1 shows the weight each problem has for each subject (i.e., problem requirement), and collaborator profile given in the lower part of the table shows the degree of expertise each collaborator has for each subject (i.e., collaborator requirement). Using the information in these profiles, we calculated correlations between subjects based on Eq. (1), which is shown in Table 2. For example, correlation between $s_1$ and $s_2$ can be calculated as follows.

$$C(s_1, s_2) = \frac{0.4 \times 0.2 + 0.1 \times 0.2}{\sqrt{0.4^2 + 0.1^2} \times \sqrt{0.2^2 + 0.2^2}} \times \frac{2}{6} = 0.2858$$

For the implementation of GA, we set the population size to 4 and the number of generations as 6 in order to simplify the illustration. Now, each step of GA can be explained as follows.

### Table 1. An example of problem and collaboration profile

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.2</td>
<td>0.5</td>
<td>0.3</td>
<td>0.7</td>
<td>0.6</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
<td>0.6</td>
<td>0.4</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
<td>0.6</td>
<td>0.5</td>
<td>0.8</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
<td>0.6</td>
<td>0.2</td>
<td>0.5</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>0.1</td>
<td>0.4</td>
<td>0.4</td>
<td>0.1</td>
<td>0.2</td>
<td>0.6</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
<td>0.2</td>
<td>0.6</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>0.15</td>
<td>0.35</td>
<td>0.25</td>
<td>0.15</td>
<td>0.15</td>
<td>0.25</td>
<td>0.15</td>
<td>0.15</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
<td>0.2</td>
<td>0.6</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

### Table 2. The correlation matrix for the example in Table 1

<table>
<thead>
<tr>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3162</td>
<td>0.1680</td>
<td>0.3237</td>
<td>0.1067</td>
<td>0.5063</td>
<td>0.4689</td>
<td>0.2109</td>
<td>0.3922</td>
</tr>
<tr>
<td>0.3162</td>
<td>1</td>
<td>0.4680</td>
<td>0.3400</td>
<td>0.1640</td>
<td>0.3750</td>
<td>0.1637</td>
<td>0.2109</td>
<td>0.3922</td>
</tr>
<tr>
<td>0.1680</td>
<td>0.4680</td>
<td>1</td>
<td>0.3000</td>
<td>0.1030</td>
<td>0.2687</td>
<td>0.3398</td>
<td>0.3333</td>
<td>0.3333</td>
</tr>
<tr>
<td>0.3237</td>
<td>0.3400</td>
<td>0.3000</td>
<td>1</td>
<td>0.2000</td>
<td>0.2000</td>
<td>0.2000</td>
<td>0.2000</td>
<td>0.2000</td>
</tr>
<tr>
<td>0.1067</td>
<td>0.1640</td>
<td>0.1030</td>
<td>0.2000</td>
<td>1</td>
<td>0.4112</td>
<td>0.2981</td>
<td>0.4002</td>
<td>0.4002</td>
</tr>
<tr>
<td>0.5063</td>
<td>0.3750</td>
<td>0.2687</td>
<td>0.2000</td>
<td>0.4112</td>
<td>1</td>
<td>0.4112</td>
<td>0.4112</td>
<td>0.4112</td>
</tr>
<tr>
<td>0.4689</td>
<td>0.1637</td>
<td>0.3398</td>
<td>0.2000</td>
<td>0.2981</td>
<td>0.4112</td>
<td>1</td>
<td>0.4112</td>
<td>0.4112</td>
</tr>
<tr>
<td>0.2109</td>
<td>0.2109</td>
<td>0.3398</td>
<td>0.2000</td>
<td>0.4002</td>
<td>0.4112</td>
<td>0.4112</td>
<td>1</td>
<td>0.4112</td>
</tr>
<tr>
<td>0.3922</td>
<td>0.3922</td>
<td>0.3333</td>
<td>0.2000</td>
<td>0.4002</td>
<td>0.4112</td>
<td>0.4112</td>
<td>0.4112</td>
<td>1</td>
</tr>
</tbody>
</table>

### Figure 3. Chromosomes in the initial population

**Step 2: Generating new chromosomes by crossover:** After constructing the initial population of a fixed number of chromosomes (i.e., 4 in our example), crossover operation is applied to two parent chromosomes randomly chosen from the population to yield new child chromosomes which are to be included in the next generation. Figure 4 shows the results of crossover operation applied to two chromosomes 1 and 3. Each chromosome was divided at two points of the chromosome (i.e., after the third and the sixth genes) into 3 segments and then we exchanged the second segment of the two chromosomes. As a result of this crossover, two new child chromosomes, 5 and 6, were generated. Note that chromosome 5 has duplicated genes (i.e., third and fourth genes) in problem 2. It means one collaborator was assigned to the same problem (e.g., $p_3$) twice, which is infeasible. Therefore, one of the duplicated genes should be randomly selected and changed into another one which is not assigned to the problem and meets the working condition requirement, as is represented in italics in Figure 4.

### Figure 4. The first application of crossover operation

**Step 3: Calculating the fitness values of chromosomes:** To evaluate the goodness of the two newly generated chromosomes, 5 and 6, the fitness values of them are calculated, using Eq. (2). The fitness values of chromosomes 5 and 6 are 0.2326 and 0.2232, respectively. By comparing the fitness values of newly generated chromosomes with those
of chromosomes in the population, 4 chromosomes 1, 3, 5, and 6 which have low fitness values were selected as parent chromosomes for the next population, as is shown in Figure 5.

Figure 5. Chromosome in the second population

Step 4: Repeating Steps 2 and 3 many times: Crossover operation (i.e., Step 2 and 3) were repeated three times, as is shown in Figure 6.

Step 5: Generating new chromosomes by mutation: Having applied crossover operator 3 times, mutation operator randomly changes the values of genes in order to lower the probability of falling into a local optimization.

![Image of repeated application of crossover operation](image)

Figure 6. Repeated application of crossover operation

Suppose mutation operator selected two chromosomes 1 and 8 and changed two genes of them (i.e., the sixth gene and the seventh gene of chromosome 1 and the fifth gene and the seventh gene of chromosome 8) into other ones (i.e., 6 into 10 and 7 into 9 in chromosome 1, and 1 into 5 and 7 into 2 in chromosome 8) to produce a new chromosomes, 11 and 12, as is shown in Figure 7. The resulting chromosomes do not violate the working condition requirement.

![Image of an example of mutation operation](image)

Figure 7. An example of mutation operation

Step 6: Calculating the fitness values of chromosomes: In the same manner as in Step 2, the fitness values of newly generated chromosomes, 11 and 12, are calculated (i.e., 0.1887 and 0.2224, respectively), and compared with those of chromosomes in the population. And then again, 4 chromosomes which have low fitness values (i.e., chromosomes 1, 8, 11, and 12) were selected as parent chromosomes for the next population, as is shown in Figure 8.

![Image of chromosomes in the fifth population](image)

Figure 8. Chromosomes in the fifth population

Steps 7: Repeating Step 2 to 6 until a stopping condition (e.g., 6 generations) is met: After repeating the process of generating new populations of chromosomes with the number of epochs being 6, the best chromosome 11, having the lowest fitness value of all the chromosomes generated until then, was selected as a sub-optimal solution to the problem, as is shown in Figure 9 (i.e., Firm requirement). Information in chromosome 11 implies that in order to solve the problems sub-optimally, collaborators 1 and 2 should be assigned to problem 1, collaborators 4, 9, and 10 to problem 2, and collaborators 8, 9, and 10 to problem 3. Also, the chromosome 11 does not violate the subject requirement, as is shown in Table 3.

5. Conclusions

This study proposed a GA-based global optimization method for assigning suitable collaborators to multiple problems considering the characteristics of problems and expertise of collaborators.
obtained from a business environment. Next, the GA-

better to carry out an experiment with a real data

account the following limitations. First, it would be

research results, however, we need to take into

example, not with a real data. For more satisfactory

optimally, but in a relatively short time.

applied GA to solve multiple complex problems sub-

collaborators to define selection criteria; 3) we

characteristics of problems and the expertise of

complex problems; 2) we took into account both the

is different from them in three aspects: 1) we focused

on collaborator selection for the solving of multiple

solution based on the problem and collaborator

GA process to the problem of finding an optimal

each subject of individual collaborators. We applied

collaborator profile shows the degree of expertise on

weight of each subject in individual problems and

problem profile shows the

We presented a method for building problem and collaborator profiles. Problem profile shows the weight of each subject in individual problems and collaborator profile shows the degree of expertise on each subject of individual collaborators. We applied GA process to the problem of finding an optimal solution based on the problem and collaborator profiles. Our approach can be extended to another situation, in which a third-party organization works as an intermediary between firms with complex problems and job-seekers having expertise required by the problems, with slight modification.

As compared with previous researches, our study is different from them in three aspects: 1) we focused on collaborator selection for the solving of multiple complex problems; 2) we took into account both the characteristics of problems and the expertise of collaborators to define selection criteria; 3) we applied GA to solve multiple complex problems sub-optimally, but in a relatively short time.

We illustrated our proposed method with an example, not with a real data. For more satisfactory research results, however, we need to take into account the following limitations. First, it would be better to carry out an experiment with a real data obtained from a business environment. Next, the GA-based optimization method tends to identify a sub-optimal solution instead of the optimal solution. Therefore, the acceptability of GA can be limited when the optimized allocation of manpower is strictly required. As such, it would be necessary to define the fitness function properly according to a problem at hand as well as the stopping condition, to get a better sub-optimal solution. Nonetheless, we believe that our approach is well suited for this kind of problem with many factors to be considered, and has the potential to produce a good (though often sub-optimal) result in less time than other complex and time consuming methods.

References