IEMS Vol. 10, No. 1, pp. 54-64, March 2011.

A Case Study of Human Resource Allocation for Effective Hotel Management

Kayoko Murakami†
School of Arts and Sciences
Shibaura Institute of Technology Saitama, 337-8570, JAPAN
Tel: +81-48-687-5072, E-mail: kayoko@shibaura-it.ac.jp

Seren Ozmehmet Tasan
Department of Industrial Engineering
Dokuz Eylul University Bornova, Izmir, 35160, TURKEY
Tel: +90-232-412-1212, E-mail: seren.ozmehmet@deu.edu.tr

Mitsuo Gen
Fuzzy Logic Systems Institute Izuka, Fukuoka, 820-0067, JAPAN
Tel: +81-948-24-2771, E-mail: mitsuogen@gmail.com

Takashi Oyabu
Graduate School of Strategic Management
Kanazawa Seiryo University Kanazawa, Ishikawa, 920-8620, JAPAN
Tel: +81-76-253-3962, E-mail: oyabu@seiryo-u.ac.jp

Received, January 17, 2011; Revised, February 13, 2011; Accepted, February 21, 2011

Abstract. The purpose of this study is to optimally allocate the human resources to tasks while minimizing the total daily human resource costs and smoothing the human resource usage. The human resource allocation problem (hRAP) under consideration contains two kinds of special constraints, *i.e.* operational precedence and skill constraints in addition to the ordinary constraints. To deal with the multiple objectives and the special constraints, first we designed this hRAP as a network problem and then proposed a Pareto multistage decision-based genetic algorithm (P-mdGA). During the evolutionary process of P-mdGA, a Pareto evaluation procedure called generalized Pareto-based scale-independent fitness function approach is used to evaluate the solutions. Additionally, in order to improve the performance of P-mdGA, we use fuzzy logic controller for fine-tuning of genetic parameters. Finally, in order to demonstrate the applicability and to evaluate the performance of the proposed approach, P-mdGA is applied to solve a case study in a hotel, where the managers usually need helpful automatic support for effectively allocating hotel staff to hotel tasks.

Keywords: Human Resource Allocation, Operational Precedence Constraint, Skill, Genetic Algorithm, Pareto Evaluation, Smoothing Resource Usage

1. INTRODUCTION

Simply, a human resource allocation problem (hRAP) is defined as a problem of allocating the human resources to tasks. During recent decades, the importance of human resource allocation has been recognized in many business fields. There are a variety of application areas of this problem such as transportation systems (airlines, railways, and buses), health care systems (emergency room and hospitals), protection and emergency services (police and fire and security services), call centers, civic service and utilities, venue management (ground operations at an airport, cargo terminals, casinos, and sporting venues), financial services, hospitality and tourism industries (hotels, tourist resorts, and restaurants), retails, and manufacturing industries (Ernst *et al.*, 2004). However, the requirements for human resource allocation vary according to the application areas. The

† : Corresponding Author

success of the system is usually attributed to how to deal with human resource management. Therefore, it is a key issue in order to increase the satisfaction and the profit of the system.

Especially in the hospitality and tourism industries, when hotels use their human resources, they need to consider appropriate and effective allocation of the resources to tasks because the hotel's success or failure depends on its human resource management. Therefore, effective human resource allocation is essential for the success of the hotel. Besides the simple allocation constraints, the real-world hRAP usually contains special constraints such as precedence and skill constraints. For precedence constraints, the operational precedence relations among tasks represent the physical relations between tasks. For example, breakfast can be served after cooking/preparing the foods. For skill constraints, the skill requirements of a task represent some special skills required to perform that task such as accounting skills. When a task requires some skills, a human resource who deals with the task should possess the skills (Ernst *et al*., 2004). If a task is appropriately allocated to a human resource, the human resource's performance level of the task will increase.

Creating an efficient human resource allocation is time-consuming and the results are often less optimal because it is usually done manually. Particularly, in hotel sector, human resource managers usually try to allocate human resources to tasks with their experience or intuition. Most of the time, they just use a full list of tasks and a short list of corresponding possible candidates, and then match the human resources to each task. Though, when the number of resources or tasks is large, it is difficult to find the best schedule by hand because numerous comparisons of the solution may exist. Since the content of hotel management requires abilities of human resources to carry out a variety of tasks efficiently and the hRAP is a NP-hard problem, the managers or human resource deployment professionals need helpful automatic support for creating successful human resource allocations.

Since Litsios (1965) first defined resources as reusable type and nonreusable type in resource allocation problems in 1965, the resource allocation problems have been receiving attentions of researchers. Initially, the researchers have proposed several traditional optimum seeking methods such as integer programming (IP) (Lin *et al*., 2000; Brusco and Jacobs, 2001), dynamic programming (DP) (Hussein and Abo-Sinna, 1995; Beasley and Cao 1998), and goal programming (GP) (Easton and Rossin, 1996; Kwak and Lee, 1997), in order to solve hRAP optimally. These traditional optimization methods, however, have been proved to be computationally inefficient since a hRAP belongs to NP-hard class optimization problems. Many researchers such as Saaty *et al.* (2007) concluded that it was difficult to solve a real-world hRAP with a traditional linear programming (LP) itself because the problem structure contained variables that could not be always quantified. Therefore, in recent years, majority of researchers started to focus their attentions to using approximation methods which do not guarantee finding the best solution. Nevertheless, they gave good acceptable solutions of a problem. Among approximation methods, simulated annealing (SA) (Brusco, 1995; Easton and Rossin, 1997), tabu search (TS) (Alvarez-Valdes *et al*., 1999; Gartner *et al*., 2001), and genetic algorithms (GA) (Nammuni *et al*., 2002; Lee *et al*., 2003; Rachmawati and Srinivasan, 2005; Yoshimura *et al*., 2006; Lin and Gen, 2008) were used by some researchers for solving hRAP.

Although all of those studies showed that approximation methods especially GA were good methods to solve hRAP, some important issues still remain unsolved in hRAP because only a few researchers have dealt with the realistic skill requirement constraints. Moreover, none of them considered the operational precedence constraints, which exist in real-world problems. Our previous study, which dealt with single objective of minimizing total human resource costs for the hRAP in a hotel, considered operational precedence relation between tasks and skill requirements of human resources (Hirano *et al*., 2008). The difficulties to seek an optimum human resource allocation come from these constraints.

In this study, we focus attention on designing an effective Pareto multistage decision-based genetic algorithm (P-mdGA) approach solving bi-objective hRAP in a hotel while considering special constraints, *i.e.*, operational precedence and skill requirement constraints. The objectives considered in this study are the minimization of the total daily human resource costs and the smoothing of the human resource usage. Additionally, the generalized Pareto-based scale-independent fitness function (gpsiff) approach is used to evaluate the solutions, and Fuzzy Logic Controller (FLC) is used for fine-tuning of genetic parameters.

The rest of this paper is structured as follows: in section 2, the definitions of the hRAP in a hotel system are presented together with the mathematical model; later in section 3, the proposed approach is introduced in detail; in section 4, the results from computational experiments on a case study are illustrated and the analysis of the data is described; lastly, in section 5, concluding remarks and future research directions are given.

2. BI-OBJECTIVE HUMAN RESOURCE ALLOCATION PROBLEM WITH PRECEDENCE AND SKILL CONSTRAINTS

The hRAP considered in this study can be stated as follows: a system consists of a set of subtasks $U = \{1, 2, \cdots, n_i\}$ categorized into tasks $V = \{1, 2, \cdots, m\}$ where each subtask has to be processed. The dummy activities s and t represent the start and the termination of the system. The processing time of subtask j in task i is denoted by p_{ij} where processing time of dummy subtasks are zero.

Additionally, the subtasks are interrelated through operational precedence constraints and skill constraints. The operational precedence constraints ensure that the subtask j is not started before all its predecessors have been finished. The skill constraints ensure that there is a possible set of skills R_{ij} which is required to perform subtask j in task i. There are K human resources in the system with a skill set of A_k to be allocated to subtasks.

In Figure 1, the conceptual model of hRAP is illustrated using the notations and indices.

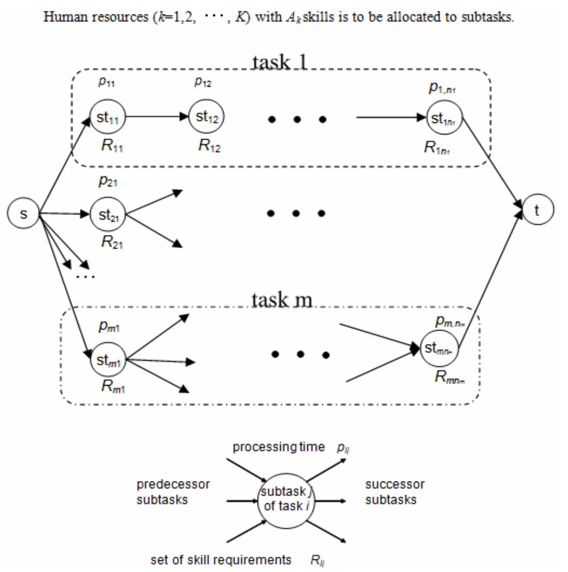

Figure 1. Conceptual hRAP model.

Traditionally, when dealing with hRAP, the most used objective function is the minimization of the total resource costs. However, smoothing of resources is also an important objective at the same time. These two objectives usually conflict with each other. Since a good hRAP solution depends on the costs of human resources and the balance of human resource usage, in this study, we consider these two objectives together in the context of a bi-objective hRAP. For the resource smoothing, we try to minimize the variance of full-time resources' usage for all activities. Since full-time resources do not work hourly like part-time resources, they need to be assigned with their maximum working hours in a day while using part-time resources less so that a total daily costs for human resources can be minimized.

For hRAP considered in this study, the following assumptions are made:

1. Task consists of a set of subtasks, which cannot be interrupted.
2. Subtask processing times are deterministic and do not differ among resources.
3. A subtask should be completed within the required time.
4. Certain subtasks require special skills.
5. An operational precedence constraint exists between subtasks and it is constant.
6. Each subtask must perform at least by predetermined number of resources.
7. Resources are available in limited quantities and they are reusable type like staff.
8. A resource can be allocated to more than one subtask as long as the subtasks do not overlap in time.
9. Resources have different employment status such as full-time and part-time.
10. Resources can only be used for a limited amount of time and cannot be used overtime.
11. Resources have different skills.
12. Costs of resources are known.
13. Costs for part-time resources are associated with hourly wages.
14. For full-time resources, costs are associated with daily wages.

In this study, the following indices, parameters, and decision parameters are used to formulate the mathematical model:

Indices

i: index of task, $i = 1, 2, \cdots, m$

j: index of subtask in task i, $j = 1, 2, \cdots, l, \cdots, n_i$ $(l \prec j)$

k: index of staff, $k = 1, 2, \cdots, K$

q: index of skill, $q = 1, 2, \cdots, Q$

Parameters

m: total number of tasks

n_i: total number of subtasks in task i

K: total number of staff

Q: total number of skills

Parameters for Subtask:

t_{ij}^S: available starting time of subtask j in task i

t_{ij}^T: available termination time of subtask j in task i

p_{ij}: processing time of subtask j in task i

r_{ijq}: skill q required by subtask j in task i

R_{ij}: the possible set of skills required by subtask j in task i; $r_{ijq} \in R_{ij}$

Parameters for Staff:

w_{ij}: number of staff required for subtask j of task i

c_k: cost of staff k per hour (day for full-time staff)

H_k: maximum hours that staff k can work

$$b_k^1 = \begin{cases} 1, & \text{if staff } k \text{ is part-time (type 1)} \\ 0, & \text{otherwise} \end{cases}$$

$$b_k^2 = \begin{cases} 1, & \text{if staff } k \text{ is full-time (type 2)} \\ 0, & \text{otherwise} \end{cases}$$

$$a_{kq} = \begin{cases} 1, & \text{if staff } k \text{ has skill of } q \\ 0, & \text{otherwise} \end{cases}$$

A_k : skill set of staff k; $a_{kq} \in A_k$

F : set of full-time staff

w_k : total working hours for full-time staff k

w : average total working hours for all full-time staff

Decision Variables

$$x_{ijkq} = \begin{cases} 1, & \text{if staff } k \text{ with skill } q \text{ is assigned} \\ & \text{to subtask } j \text{ of task } i \\ 0, & \text{otherwise} \end{cases}$$

t_{ij}: actual starting time of subtask j in task i

The mathematical model for the bi-objective md-GA can be stated as follows:

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{k=1}^{K}\sum_{q=1}^{Q} b_k^1 p_{ij} c_k x_{ijkq} + \sum_{k=1}^{K} c_k \min\left\{ \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{q=1}^{Q} b_k^2 x_{ijkq},\ 1 \right\} \tag{1}$$

$$\min \quad V = \frac{1}{\sum_{k=1}^{K} b_k^2} \sum_{k \in F} \left(w_k - w\right)^2$$

$$w_k = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{q=1}^{Q} b_k^2 p_{ij} x_{ijkq},\ k \in F \tag{2}$$

$$w = \frac{1}{\sum_{k=1}^{K} b_k^2} \sum_{k \in F} w_k$$

$$\text{s. t.} \quad t_{il} + p_{il} \le t_{ij}^S,\ \forall i, j, l \tag{3}$$

$$\sum_{j=1}^{n_i}\sum_{k=1}^{K}\sum_{q=1}^{Q} x_{ijkq} \le n_i,\ \forall i \tag{4}$$

$$t_{ij} \ge t_{ij}^S,\ \forall i,\ j \tag{5}$$

$$t_{ij} + p_{ij} \le t_{ij}^T,\ \forall i, j \tag{6}$$

$$\sum_{l \in L}\sum_{j=1}^{n_l}\sum_{q=1}^{Q} x_{ljkq} \le 1,\ \forall k,\ L = \left\{ l \mid t_{il}^S \le t_{il} + \alpha \le t_{il}^T \right\} \tag{7}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{q=1}^{Q} p_{ij} x_{ijkq} \le H_k,\ \forall k \tag{8}$$

$$\sum_{k=1}^{K}\sum_{r_{ijq} \in R_{ij}} a_{kr_{ijq}} x_{ijkr_{ijq}} = w_{ij},\ \forall i, j \tag{9}$$

$$x_{ijkq} \in \{0,\ 1\},\ \forall i, j, k, q \tag{10}$$

$$t_{ij} \ge 0,\ \forall i, j \tag{11}$$

In this mathematical formulation, the objective function (1) represents the minimization of total staff costs in a daily schedule. The cost for a full-time resource is calculated with daily wage while the cost for a part-time resource is calculated with hourly wage. The objective function (2) represents the smoothing of full-time resource usage, where the minimizing the variance of full-time resource usage is considered because of the characteristics of their wage system. Once a full-time resource is assigned, his or her daily wage is added. Constraints (3)-(11) are the constraints for the feasibility. Constraint (3) represents the precedence constraints between each subtask. Constraint (4) ensures that each subtask in a task must be done by one staff member. Constraint (5) represents the constraint for the start time for each subtask. Constraint (6) ensures that each subtask is completed within the required time. Constraint (7) shows that each staff member deals with only one subtask at a time. Constraint (8) represents working-hour limitation for each resource. Constraint (9) represents skill requirements for each subtask where it ensures that each subtask is done by one staff member with required skills.

3. PARETO MULTISTAGE-BASED GENETIC ALGORITHM

In this study, to solve the bi-objective hRAP with special constraints, a Pareto multistage decision-based genetic algorithm approach (P-mdGA) is proposed. In the P-mdGA, first the hRAP model is constructed and divided into several stages. For hRAP, the subtasks are considered as the stages. The order of the stages represents the sequence of subtasks. Later, possible states of each stage are determined. For hRAP, the states are represented by human resources. To deal with the multiobjective concept, generalized Pareto-based scale-independent fitness function (gpsiff) approach is adopted and to adjust the rate of crossover and mutation, fuzzy logic controller is adopted. Let $P(t)$ be parents and $C(t)$ be offspring in current generation t. The overall procedure of P-mdGA can be stated as follows:

procedure: P- mdGA for bi-objective hRAP
input: hRAP data, GA parameters
output: Pareto optimal solutions E
begin
    $t \leftarrow 0$;
    initialize $P(t)$ by *priority-based* and *resource permutation encoding routines*;
    calculate objectives $f_1(P(t))$ and $f_2(P(t))$ by *priority-based* and *resource permutation decoding routines*;
    create Pareto $E(P(t))$;
    evaluate *eval*$(P(t))$ by *gpsiff routine*;
    while (not terminating condition**) do**
        create $C(t)$ from $P(t)$ by *position-based crossover routine*;
        create $C(t)$ from $P(t)$ by *swap mutation routine*;
        calculate objectives $f_1(C(t))$ and $f_2(C(t))$ by *priority-based* and *resource permutation decoding routines*;
        update Pareto $E(P(t),C(t))$;
        evaluate *eval*$(P(t),C(t))$ by *gpsiff routine*;
        if $t > u$ then
            fine-tuning p_M and p_C by FLC;
        select $P(t+1)$ from $P(t)$ and $C(t)$ by *modified elitist selection routine*;
        $t \leftarrow t+1$;
    end
    output Pareto optimal solutions $E(P, C)$;
end

3.1 Genetic Representation

In order to apply P-mdGA to hRAP effectively, converting the problem model into a chromosome representation is the primary concern (Gen *et al.*, 2008). The P-mdGA consists of three phases: creating a feasible subtask sequence, assigning subtasks to resources, and designing a schedule. For hRAP, the individual is composed of two chromosomes. Figure 2 represents a feasible solution (individual) showing both a subtask sequence and a resource assignment.

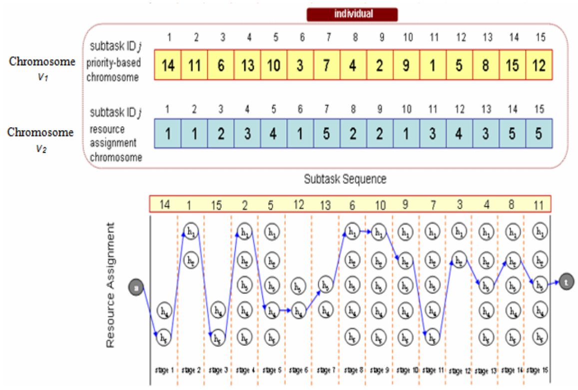

Figure 2. Genetic representation of an individual in P-mdGA.

Phase 1: Creating a feasible subtask sequence

step 1.1: Generate a random priority to each subtask in a task by using an encoding procedure

In this step, the priority-based encoding method pro-posed by Gen and Cheng (2000), which is an indirect representation scheme, is used. In this method, the position of a gene represents a subtask node and the value of the gene represents the priority of the subtask node for constructing a schedule among candidates. The procedure for the priority-based encoding is shown in Figure 3. In Figure 2, an example output of priority-based chromosome is shown as chromosome v_1. As an initialization, this encoding method is used. After initialization, genetic operators, which are crossover and mutation, are applied in order to make solution candidates. These genetic operators will be explained in detail in the following section.

procedure: priority-based encoding
input: number of subtasks n
output: chromosome $v_1()$
begin
  for $j = 1$ to n
    $v_1(j) \leftarrow (1, n)$;
  end
  for $j = 1$ to $n/2$
    $j \leftarrow$ random$(1, n)$;
    $k \leftarrow$ random$(1, n)$;
    if $j \neq k$ then
      swap $\{v_1(j), v_1(k)\}$;
  end
  output chromosome $v_1()$
end

Figure 3. Priority-based encoding procedure.

step 1.2: Decode a feasible subtask sequence T_S that satisfies the operational precedence constraint

After generating priority-based chromosomes in step 1.1, the priorities of each subtask are used to create a feasible subtask sequence that satisfies the precedence constraint in the model. Figure 4 presents the priority-based decoding procedure for creating a feasible subtask sequence. In Figure 2, a feasible subtask sequence is shown.

procedure: priority-based decoding
input: number of subtasks n, chromosome $v_1()$
output: a subtask sequence T_S
begin
  $\bar{s} \leftarrow \varnothing$, $T_S \leftarrow \varnothing$;
  $n \leftarrow 0$, $j \leftarrow 0$;
  while $(j \leq n)$ **do**
    $\bar{s} \leftarrow \mathrm{Suc}(j)$;
    $j^* \leftarrow \arg\max\{v_1(j) \mid j \in \bar{s}\}$;
    $\bar{s} \leftarrow \bar{s} \setminus j^*$;
    $T_S \leftarrow T_S \cup j^*$;
    $j \leftarrow j^*$;
  end
  output a feasible subtask sequence T_S
end

Figure 4. Priority-based decoding procedure.

Phase 2: Assigning subtasks to resources

step 2.1: Assign each subtask to resources by using resource permutation coding procedure

In this phase, the assignments of subtask to resources are formed using the permutation coding procedure while satisfying the resource skill requirement for each subtask. Figure 5 shows the resource permutation encoding. In Figure 2, the resource assignments are shown as chromosome v_2.

procedure: resource permutation encoding
input: number of staff K, number of states at stage j, n_j
output: chromosome $v_2()$
begin
  for $j = 1$ to K
    $v_2(j) \leftarrow 0$;
  end
  for $j = 1$ to J
    $v_2(j) \leftarrow$ random$[1, n_j]$;
  end
  output chromosome $v_2()$
end

Figure 5. Resource permutation encoding procedure.

step 2.2: Obtain a feasible assignment according to the subtask sequence found in step 1.2 and the resource assignment found in step 2.1

Using the task sequence and the resource permutation encoding, a feasible solution is obtained using the resource permutation decoding procedure in Figure 6. Figure 2 illustrates the feasible subtask sequence and its resource assignment. After this, next generation will be produced by using the selection operator, which is discussed in the following section.

Phase 3: Designing a schedule
step 3.1: Create a schedule S using the resource assignment to each subtask found in step 2.2

According to the feasible resource assignment, the schedule using the starting and the termination times of subtask can be constructed as follows:

S = {(st_{11}, h_1: 4:00~6:00), (st_{12}, h_1: 7:00~8:00), (st_{13}, h_2: 15:00~18:00), (st_{14}, h_3: 18:00~19:00), (st_{21}, h_4: 6:00~8:00), (st_{22}, h_1: 8:00~10:00), (st_{23}, h_5: 15:00~17:00), (st_{24}, h_2: 18:00~20:00), (st_{31}, h_2: 10:00~12:00), (st_{32}, h_1: 10:00 ~12:00), (st_{33}, h_3: 21:00~22:00), (st_{41}, h_4: 8:00~12:00), (st_{42}, h_3: 12:00~16:00), (st_{51}, h_5: 7:00~10:00), (st_{52}, h_5: 12:00~15:00)}

step 3.2: Draw two Gantt charts for this schedule from two points of view: staff schedule and subtask schedule

Final step is to effectively visualize the schedule using two Gantt charts showing the schedule from the human resource and subtask point of view separately.

```
procedure: resource permutation decoding
input: chromosome v2(), subtask sequence TS, nu
       mber of staff K
output: schedule S
begin
    S ← ∅;
    J ← 1;
    while (j ≤ J) do
        r ← v2 ( j);
        nj ← Ts ( j);
        S ← S ∪{(nj, kr)};
        j ← j+1;
    end
    output schedule S
end
```

Figure 6. Resource permutation decoding procedure.

3.2 Genetic Operators

In the proposed P-mdGA, the position-based crossover, swap mutation, and modified elitist selection are used as genetic operators for priority-based chromosome v_1.

By using the position-based crossover, some genes are taken from one parent at random, and they fill the same positions in offspring (Gen and Cheng, 2000). Then, the vacuum positions in offspring are filled with genes from the other parent by a left-to-right scan. The position-based crossover can keep the characteristics of gene order in parent chromosomes. However, this crossover does not keep much of the characteristics of parents like one-cut crossover.

In the swap mutation operator, two positions are selected at random and their contents are swapped in order to produce random changes in various chromosomes spontaneously (Gen and Cheng, 2000). The swap mutation operator makes the chromosomes a big change in terms of gene position by swapping only two genes. Since the position-based crossover does not change the characteristics of gene position much, this swap mutation is used.

The modified elitist selection preserves the best chromosome in the next generation and overcomes the stochastic errors of sampling (Gen and Cheng, 2000). If the best individual in the current generation is not reproduced into the new generation, one individual is randomly removed from the new population and the best one from the current population is added to the new population. This selection gives GA better search ability in the solution areas without being stuck at local optima. Usually, the elitist selection makes the search area get smaller, which means less variety, by choosing the elitist chromosome. However, this modified version chooses the good one among the best. While preserving the best chromosome in the next generation, this selection gives more diversification to the solution area.

3.3 Pareto Evaluation with gpsiff Approach

In our bi-objective hRAP model, we have used the generalized Pareto-based scale-independent fitness function (gpsiff) approach for the evaluation of the Pareto solutions during P-mdGA. The gpsiff approach makes the use of Pareto dominance relationship to evaluate individuals using a single measure of performance (Ho *et al.*, 2004). It uses a pure Pareto-ranking fitness assignment strategy, which differs from the traditional Pareto-ranking methods. Let the fitness value of an individual x be a tournament-like score obtained from all participant individuals by the following eq. (12) where p is the number of individuals which can be dominated by the individual x, and q is the number of individuals which can dominate the individual x in the objective space. Generally, a constant c can be optionally added in the fitness function to make fitness values positive. In this study, c is the number of all participant individuals.

$$eval(x_i) = p(x_i) - q(x_i) + c,\ i = 1, 2, \cdots, popSize \tag{12}$$

where
$$p(x_i) = \left\{ |x_j| \;\middle|\; \begin{matrix} f_k(x_j) \ge f_k(x_i),\ \forall\, k, j, \\ {\scriptstyle x_i \ne x_j} \\ (k = 1, 2, \cdots, q,\ j = 1, 2, \cdots, popSize) \end{matrix} \right\}$$

$$q(x_i) = \left\{ |x_j| \;\middle|\; \begin{matrix} f_k(x_i) \ge f_k(x_j),\ \forall\, k, j, \\ {\scriptstyle x_i \ne x_j} \\ (k = 1, 2, \cdots, q,\ j = 1, 2, \cdots, popSize) \end{matrix} \right\}$$

3.4 Genetic Parameter Tuning with FLC

For the regulation of GA parameters, fuzzy logic controller (FLC), which improves the exploration performance of GA by tuning parameters automatically for each generation and assigning better conditions for exploring optimal solution, has been proved to be very useful. In this research, for the fine-tuning of genetic parameters, Wang *et al.*'s FLC concept is used (1997). This concept consists of two FLCs, *i.e.*, crossover FLC and mutation FLC, which are implemented independently. Using these two FLCs, we adaptively adjust the crossover probability and mutation probability during the optimization process as shown in Figure 7.

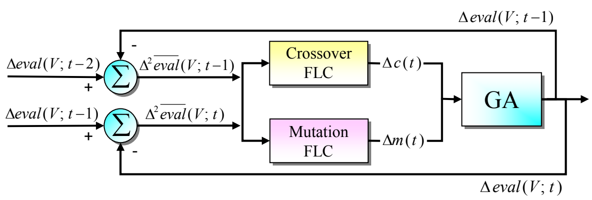

Figure 7. Structure of fuzzy logic controller.

4. CASE STUDY IN JAPANESE HOTEL SECTOR

In order to show the applicability of the proposed approach, we solved a case study with the data modified from real data collected from several Japanese-style hotels, which are particularly called as ryokan. In a ryokan, guests can experience the elements of Japanese culture and customs. Since almost all ryokans provide breakfast/dinner, and also feature common or private hot-spring bathing areas in addition to bathrooms, services and their associated tasks are quite different from normal hotel systems. Additionally, staff can do all the services if he/she has the required service skills. Usually, the full-time staff member working for the ryokan lives within the hotel premises so that the working hours can be more flexible than staff members working for the ordinary hotel systems.

In this case study, 5 staff members are to be allocated to 15 subtasks. Each subtask requires skills which are defined in Table 1. The information related with subtasks is shown in Table 2. The set of required skills to perform each subtask is shown in the last column of this table. For example in order to perform subtask st13, which requires chef skills, the required skill is skill 1 or skill 2.

The operational precedence relations are shown in Figure 8. Table 3 shows the list of staff members. Each staff member works under different employment status, *e.g.* staffs 1 and 2 have full-time and part-time employment status, respectively.

Table 1. The list of skills in the case study.

Skill q	Skills
1	chef with more than 10 years experience
2	chef with less than 10 years experience
3	foreign language
4	accountant with more than 10 years experience
5	accountant with less than 10 years experience

Table 2. The list of subtasks in the case study.

Task category i	Subtask st_{ij}	Subtask Index	Processing time (h) p_{ij}	Earliest starting time t_{ij}^{S}	Latest finishing time t_{ij}^{T}	Successor subtask	Possible set of skills R_{ij}
1: kitchen	st_{11}: breakfast	1	2	4 : 00	8 : 00	st_{12}, st_{21}	$R_{11} = \{1, 2\}$
	st_{12}: dish and cleaning (b)	2	1	7 : 00	10 : 00	st_{13}	$R_{12} = \{1, 2, 3, 4, 5\}$
	st_{13}: dinner	3	3	15 : 00	20 : 00	st_{14}, st_{24}	$R_{13} = \{1, 2\}$
	st_{14}: dish and cleaning (d)	4	1	18 : 00	22 : 00	t	$R_{14}=\{1, 2, 3, 4, 5\}$
2: serving	st_{21}: breakfast serving	5	2	6 : 00	9 : 00	st_{22}	$R_{21} = \{1, 2, 3, 4, 5\}$
	st_{22}: check-out	6	2	7 : 00	11 : 00	st_{31}, st_{32}	$R_{22} = \{1, 2, 3, 4, 5\}$
	st_{23}: check-in	7	2	14 : 00	21 : 00	st_{13}	$R_{23} = \{1, 2, 3, 4, 5\}$
	st_{24}: dinner serving	8	2	18 : 00	21 : 00	st_{33}	$R_{24} = \{1, 2, 3, 4, 5\}$
3: cleaning	st_{31}: rooms and hallway	9	2	9 : 00	14 : 00	st_{23}	$R_{31} = \{1, 2, 3, 4, 5\}$
	st_{32}: laundry	10	2	9 : 00	15 : 00	t	$R_{32} = \{1, 2, 3, 4, 5\}$
	st_{33}: bath	11	1	21 : 00	0 : 00	t	$R_{33} = \{1, 2, 3, 4, 5\}$
4: front	st_{41}: morning	12	4	8 : 00	13 : 00	st_{42}	$R_{41} = \{3\}$
	st_{42}: afternoon	13	4	12 : 00	17 : 00	t	$R_{42} = \{3\}$
5: management	st_{51}: accountant morning	14	3	7 : 00	13 : 00	st_{12}	$R_{51} = \{4, 5\}$
	st_{52}: accountant afternoon	15	3	12 : 00	19 : 00	t	$R_{52} = \{4, 5\}$

Table 3. The list of staff members in the case study.

Staff h_k	Staff (index) k	Skill set(experience) A_k	Employment status (part-time $b_k^1 = 1$, full-time $b_k^2 = 1$)	Staff wage (hourly or daily) c_k
h_1	1	{1}: chef(30 years)	full	24800(daily)
h_2	2	{2}: chef(3 years)	part	1000(hourly)
h_3	3	{3}: language	part	1300(hourly)
h_4	4	{3, 5}: language and accountant(7 years)	full	14400(daily)
h_5	5	{4}: accountant(12 years)	full	16000(daily)

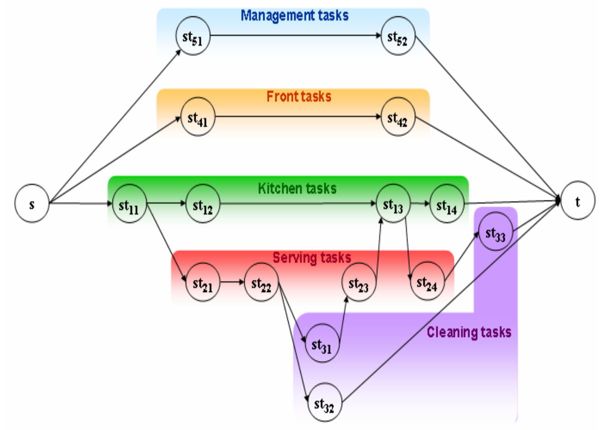

Figure 8. Operational precedence relations in the case study.

To solve this case study, we have applied the proposed P-mdGA approach by programming in C++ in PC-configuration of 2.60GHz Pentium4 with 1GB RAM. We ran the program 10 times. During the computations, the experimental conditions are constructed as follows:

population size: *popSize* = 140
terminating conditions (*maxGen* or *conLimit*)
maximum number of generations: *maxGen* = 100
convergence Limit: *conLimit* = 30
crossover rate: $p_C = 0.7$
mutation rate: $p_M = 0.02$

If FLC is not applied to the proposed P-mdGA approach, the values of p_M and p_C are kept constant as the generation is increased. On the other hand, in P-mdGA with FLC, the values of p_M and p_C are adaptively regulated by FLC. When we compared the CPU time between P-mdGA without FLC and P-mdGA with FLC, the latter approach performed with shorter time. Additionally, we compared the performance of the proposed P-mdGA using gpsiff and P-mdGA using adaptive weight approach (awa) (Gen *et al*., 2008).

To compare the results for Pareto evaluations, we have used 3 performance measures, *i.e.*, number of obtained solutions $|S_j|$ that counts the number of obtained solution set, ratio of nondominated solutions $R_{NDS}(S_j)$ that counts the number of solutions which are members of the reference solution set, and average distance $D1_R(S_j)$ that calculates the closeness of an obtained solution set from the reference solution set.

The number of obtained solutions found by P-mdGA using gpsiff and awa are 105 and 88, respectively. According to number of obtained solutions, it can be stated that P-mdGA performs better with gpsiff than awa.

The ratio of nondominated solutions is calculated by eq. (13) where $\boldsymbol{r} \prec \boldsymbol{x}$ means that the solution $\boldsymbol{x}$ is dominated by the solution $\boldsymbol{r}$. A ratio of nondominated solutions $R_{NDS}(S_j) = 1$ means all solutions are members of the Pareto-optimal set S^*, and a $R_{NDS}(S_j) = 0$ means no solution is a member of the S^*. Although the number of obtained solutions $|S_j|$ is large, if the ratio of nondominated solutions $R_{NDS}(S_j)$ is 0, it may be the worst result. The difficulty with the above measures is that although a member of S_j is Pareto-optimal, if that solution does not exist in S^*, it may not be counted in $R_{NDS}(S_j)$ as a non Pareto-optimal solution. Thus, it is essential that a large set for S^* is necessary in the above equation. The ratio of nondominated solutions found by P-mdGA using gpsiff and awa are 0.7 and 0.4, respectively. According to ratio of nondominated solutions, it can be stated that P-mdGA with gpsiff can produce better solutions than P-mdGA with awa.

$$R_{NDS}(S_j) = \frac{\left| S_j - \left\{ \boldsymbol{x} \in S_j \,\middle|\, \exists \boldsymbol{r} \in S^* : \boldsymbol{r} \prec \boldsymbol{x} \right\} \right|}{|S_j|} \tag{13}$$

To calculate the average distance, we define S_j as a solution set (j = 1, 2) for each algorithm, *i.e.*, P-mdGA with gpsiff and awa. To find a reference solution set S^*, the following parameter setting is used;

Population size: *popSize* = 1000
Crossover probability: $p_C = 0.8$
Mutation probability: $p_M = 0.3$
Terminating condition:
Maximum number of generations: *maxGen* = 10000

The average distance $D1_R(S_j)$ is calculated by the eq. (14) to find an average distance of the solutions of S_j from S^* where d_{rx} is the distance between a current solution $\boldsymbol{x}$ and a reference solution $\boldsymbol{r}$ in the 2-dimensional normalized objective space.

$$\mathrm{D1_R}(S_j) = \frac{1}{|S^*|} \sum_{r \in S^*} \min\{d_{rx} \,|\, \boldsymbol{x} \in S_j\} \tag{14}$$

$$\text{where} \qquad d_{rx} = \sqrt{\sum_{q=1}^{2} \left(f_q(\boldsymbol{r}) - f_q(\boldsymbol{x}) \right)^2}$$

The smaller value of $\mathrm{D1_R}(S_j)$ means the solution set S_j is better. This measure calculates the closeness of a solution set S_j from the set S^*. The average distance found by P-mdGA using gpsiff and awa are 2.31 and 4.16, respectively. The average distance values show the better performance of P-mdGA with gpsiff.

Based on the three performance measures found for gpsiff and awa, we can state that the performance of P-mdGA can be improved by using gpsiff. For this bi-objective hRAP, there is a set of solutions that cannot simply be compared with each other. However, in a real-world, the human resource managers are asked to select one of the solutions as the best compromised solution. This means that the best compromised solution is dependent on the subjective preference of the manager.

For this case, we can employ the factor weight method, which uses factor weights for cost and smoothing objectives respectively as the preferences of manager. For factor weights 0.5 and 0.5, the best compromised solution is illustrated in Fig. 9. For this best compromised solution, the total staff costs found to be 243,600yen. In addition, Gantt chart of the best compromised solution is shown in Fig.10.

Based on the computational experiment, the following statements can be made about the proposed P-mdGA approach; P-mdGA with FLC has proven to be an efficient solution algorithm for solving hRAP with operational precedence and skill constraints. Additionally, gpsiff can improve the performance of the proposed P-mdGA approach.

When the human resource managers try to make a better schedule considering minimizing the human resource costs, their staff's skills, or their working time preferences, it takes much time to create the compromised schedule. For example, the managers in the ryokans we collected the data need 20 minutes to deal with 5 staff members, or 1 hour for 20 staff members. Comparing the scheduling time by hand with that by automatic support, it is clear that the automatic support is helpful.

5. CONCLUSION

Human resources like hotel staffs are the first point of contact between the customers and the hotel. Especially in hotels, some services like greeting guests cannot be substituted by technology. Therefore, the effec-

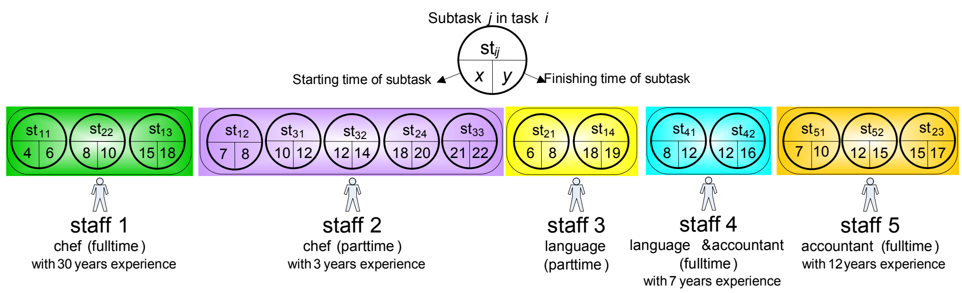

Figure 9. Allocation of staff to subtasks in the best compromised solution found by P-mdGA with gpsiff.

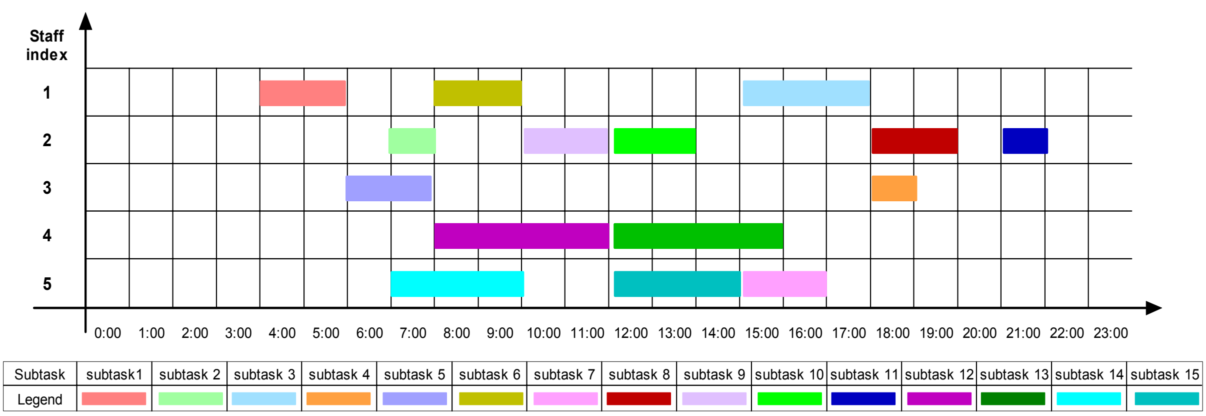

Figure 10. Gantt chart illustration of the best compromised solution found by P-mdGA with gpsiff.

tive human resource allocation plays an important role in the success of the hotel systems. In hotels, hRAP involves the allocation of hotel staff to service tasks, where staff can be used for a limited amount of time due to regulations in labor standards law. In order to create successful human resource allocations, the human resource deployment professionals need an automatic hRAP solution system especially when the problem is big-sized and/or complex.

In this study, we proposed an effective P-mdGA approach for solving the bi-objective hRAP with operational precedence and skill constraints. Traditionally, many researchers who dealt with this kind of problem either simplified or ignored the operational precedence, which exists between tasks, even though the issue of operational precedence naturally comes up when considering the work in a hotel. During evolutionary process of P-mdGA, gpsiff approach is used to deal with the bi-objective hRAP. The proposed P-mdGA has five main advantages. The first advantage is the appropriate usage of operational precedence relations and skill requirements in multistage concept of hRAP with two chromosomes, *i.e.* priority-based and resource permutation-based chromosomes. The second advantage is the usage of priority-based chromosome representation, which tries to select the activities according to their priorities and schedule them at its earliest operational precedence. The third advantage is the usage of resource permutation-based chromosome, which is used to represent stages, *i.e.* possible staff allocation solutions particularly using skill requirements of the subtasks and staff skills together. The fourth advantage is the usage of FLC, which is used for auto-tuning of P-mdGA parameters and assigning better conditions to explore the optimal solution. Finally, the fifth advantage is the usage of gpsiff, which is used for evaluating Pareto solutions. Based on the case study we had, it can be stated that the proposed P-mdGA with gpsiff approach has proven to be an efficient solution algorithm.

For further research, we have to consider the allocation with more staff members to more tasks to match with by using a real-world data since this study dealt with modified data with a small number of human resources and tasks. Also, operational precedence relations between subtasks should be reviewed carefully such as cleaning tasks to be divided into more detailed subtasks.

Although the automatic support is needed to help the human resource managers create a better schedule, their experience and intuition are still necessary to make a final decision. Therefore, it should be better to use both manual and automatic ways to deal with human resource allocation.

REFERENCES

Alvarez-Valdes, R., Crespo, E., and Tamarit, J. M. (1999), Labour scheduling at an airport refueling installation, *Journal of the Operational Research Society*, **50**, 211-218.

Beasley, J. E. and Cao, B. (1998), A dynamic programming based algorithm for the crew scheduling problem, *Computers and Operations Research*, **25**, 567-582.

Brusco, M. J. and Jacobs, L. W. (2001), Starting-time decisions in labor tour scheduling: An experimental analysis and case study, *European Journal of Operational Research*, **131**, 459-475.

Brusco, M. J., Jacobs, L. W., Bongiorno, R. J., Lyons, D. V., and Tang, B. (1995), Improving personnel scheduling at airline stations, *Operations Research*, **43**, 741-751.

Easton, F. F. and Rossin, D. F. (1996), A stochastic goal program for employee scheduling, *Decision Sciences*, **27**, 541-568.

Easton, F. F. and Rossin, D. F. (1997), Overtime schedules for full-time service workers, *Omega*, **25**, 285-299.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004), Staff scheduling and rostering: A review of applications, methods and models, *European Journal of Operational Research*, **153**, 3-27.

Gartner, J., Musliu, N., and Slany, W. (2001), Rota: A research project on algorithms for workforce scheduling and shift design optimization, *AI Communications*, **14**, 83-92.

Gen, M. and Cheng, R. (2000), *Genetic Algorithm and Engineering Optimization*, John Wiley and Sons, NY.

Gen, M. and Cheng, R., and Lin, L. (2008), *Network Models and Optimization*: *Multiobjective Genetic Algorithm Approach*, Springer.

Hirano, K., Tasan, S. O., Gen, M., and Oyabu, T. (2008), Skill-based resource allocation problem by multistage decision-based genetic algorithm, *Proceedings, Asia Conference on Intelligent Manufacturing and Logistics Systems*, 463-471.

Ho, S.-Y., Shu, L.-S., and Chen, J.-H. (2004), Intelligent evolutionary algorithms for large parameter optimization problems, *IEEE Transaction on Evolutionary Computation*, **8**, 522–541.

Hussein, M. L. and Abo-Sinna, M. A. (1995), A fuzzy dynamic approach to the multicriterion resource allocation problem, *Fuzzy Sets and Systems*, **69**, 115-124.

Kwak, N. K. and Lee, C. (1997), A linear goal programming model for human resource allocation in a health-care organization, *Journal of Medical Systems*, **21**, 129-140.

Lee, Z. J., Su, S. F., Lee, C. Y., and Hung, Y. S. (2003), A heuristic genetic algorithm for solving resource allocation problems, *Knowledge and Information Systems*, **5**, 503-511.

Lin, C. K. Y., Lai, K. F., and Hung, S. L. (2000), Development of a workforce management system for a customer hotline service, *Computers and Operations Research*, **27**, 987-1004.

Lin, C. M. and Gen, M. (2008), Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm, *Expert Systems with Applications*, **34**, 2480-2490.

Litsios, S. (1965), A resource allocation problem, *Operations Research*, **13**, 960-988.

Nammuni, K., Levine, J., and Kingston, J. (2002), Skill-based resource allocation using genetic algorithms and ontologies, *Proceedings*, *International Workshop on Intelligent Knowledge Management Techniques*, IOS Press, Amsterdam.

Rachmawati, L. and Srinivasan, D. (2005), A hybrid fuzzy evolutionary algorithm for a multi-objective resource allocation problem, *Proceedings*, *5th International Conference on Hybrid Intelligent Systems*, 55-60.

Saaty, T. L., Peniwati, K., and Shang, J. S. (2007), The analytic hierarchy process and human resource allocation: Half the story, *Mathematical and Computer Modeling*, **46**, 1041-1053.

Wang, P. Y., Wang, G. S., and Hu, Z. G. (1997), Speeding up the search process of genetic algorithm by fuzzy logic, *European Congress on Intelligent Techniques and Soft Computing*, 665-671.

Yoshimura, M., Fujimi, Y., Izui, K., and Nishiwaki, S. (2006), Decision-making support system for human resource allocation in product development projects, *International Journal of Production Research*, **44**, 831-848.