

Article

Joint Requirement of Two Multi-skill Resource Types in Multi-period Multi-site Assignment Problems

Siravit Swangnop and Paveena Chaovalitwongse*

Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330, Thailand

*E-mail: paveena.c@chula.ac.th

Abstract. A classic assignment problem determines how to assign resources to tasks in the best possible way. Over the past years, the classic assignment problem has been extended and more complicated assignment models have been proposed. A multi-period multi-site assignment problem is one extension of the classic assignment problem. The number of sites and periods is increased to more than one and the decision is extended to consider assigning resources to site while concerning tasks in each site and period. Most multi-period multi-site assignment models do not concern joint resources for operation; however, in some real-life problems, there is a case in which joint resources for performing tasks are required. In this study, we consider joint requirement of two multi-skill resource types in the multi-period multi-site assignment problem and propose the mathematical model and a heuristic approach. The developed heuristic is comprised of two phases. An initial solution is generated by CPLEX in the first phase. Then the solution is improved in the second phase using proposed heuristic. This heuristic aims to allocate resources to better sites and assign resources to suitable tasks. In the computational experiment, the number of resources and the ratio of tasks requiring joint resources are varied. The result shows that the complexity of the problem highly depends on the ratio of task requiring joint resources. When all parameters are fixed except for the number of resources, there is only one range of the number of resources that makes the problem complex. For the efficiency of the algorithm, the developed heuristic can find good solutions in a short time in all ranges of the number of resources in all test problems, and optimal gaps slightly increase when the ratio of tasks requiring joint resources increases (average optimal gap of all test problems is 7.25%).

Keywords: Assignment problem, multi-skill resource, joint requirement.

ENGINEERING JOURNAL Volume 19 Issue 1

Received 24 February 2014

Accepted 11 June 2014

Published 30 January 2015

Online at <http://www.engj.org/>

DOI:10.4186/ej.2015.19.1.51

1. Introduction

A classic assignment problem is a problem of assigning each task to a different resource and each resource to a specific task to minimize the total cost [1]. Over the past years, the classic assignment problem has been extended and more complicated assignment models have been proposed, for example, complicated resources and tasks [2-5], complicated objective functions [6, 7] and more dimensions of decision variable [8-10].

A multi-period multi-site assignment problem is one extension of the classic assignment problem. The number of sites and periods is increased to more than one and the decision is extended to consider assigning resources to site while concerning tasks in each site and period. Decisions in some models are not only allocating resources to site but also assigning resources to tasks or shifts [11, 12]. This kind of assignment problem is widely found in the problem of emergency resource allocation, which is a problem of allocating multiple resources from emergency depots to disaster sites [13-15], and the problem of health staff scheduling, which is a problem of allocating or assigning doctors or nurses to shifts, wards and hospitals [11, 12, 16-23].

There are many types of resource in both problems. However, for health staff scheduling problems, most of the research considers only one type of resource in their models, which is either doctor or nurse. Goyal and Yadav [16], Trivedi and Warner [22], Gutjahr and Rauner [11], Aickelin and Dowsland [18], Dowsland [19], Burke, Cowling *et al.* [20] and Tsai and Li [21] developed nurse scheduling models whereas Carter and Lapierre [17], Costa Filho, Rivera Rocha *et al.* [12] and Goyal and Yadav [23] proposed doctor scheduling models. These resources are planned separately to fulfill demands. Similarly, although most models in the problem of emergency resource allocation consider many types of resource, for example, models of Zhang *et al.* [24], Ozdamar *et al.* [14] and Tzeng *et al.* [15] proposing the model which classifies resources into multiple groups or types, all resource types are considered separately and joint of resources for performing tasks is not concerned. Mostly, resources in these models are divided into many types and demands in each site are classified separately by the resource type. The decision is to allocate resources to fulfill the demand of each resource type.

In some real-life problems, there is a case in which joint of resources for performing tasks is required and joint requirement cannot be neglected. In the problem of health resources planning in clinic networks, their resources are divided into two types, which are doctors and nurses, and their tasks or treatments require the joining of two resource types for operation. There are also many working sites and planning horizon is divided into many periods. The planner has to decide where and what task their resources should be assigned to maximize the total profit. Considering each resource type separately may not be suitable for this case.

In this study, we consider joint requirement of two multi-skill resource types in the multi-period multi-site assignment problem and propose the mathematical model and heuristic. The remaining parts of this paper are organized as follows. In section 2, problem description and mathematical model are presented. Heuristic and computational studies are described in sections 3 and 4, respectively. Finally, the conclusion and future work are discussed in section 5.

2. Model Description

In previous studies, objective functions, decision variables and constraints of multi-period multi-site assignment problem are likely to vary depending on the related applications.

In emergency resource allocation problem, sites refer to emergency depots, public areas, hospitals or disaster sites; resources represent medical supplies, equipment or staff. The decision in most models is to allocate resources to disaster sites as soon as possible while concerning operation cost and demands in each site. Thus, the decision variable is the number of resources allocated from emergency depots to disaster sites [13-15, 25]. Some models are also concerned with the transportation of resource or vehicle routing so they also include the number of vehicles required for transporting resources between nodes into the decision [14]. Planning horizon are always multiple periods [13-15, 25]. Conditions or constraints in most research are related to the limitation of the available resources, the equilibrium of the demand and supply, the balance of the resource's flow and the capacity of the vehicles. The objectives mostly found in this application are to minimize the respond time to disaster sites [13, 15], minimize fatalities in the search and

rescue period [26], minimize unsatisfied demands [14], minimize operation cost [15] and maximize fairness of resource distribution [15].

In the problem of health staff scheduling, sites are wards, departments or hospitals, and resources represent doctors or nurses. The decision in each model depends on the scope of the problem, which is to allocate/assign doctors or nurses to shifts in a ward [16-21], to wards in a hospital [22] or to hospitals in a considered area [11, 12, 23]. Planning horizon can be one shift in advance [22] or many time periods [11, 12, 16-21]. The constraints in this problem are often divided into two groups: hard and soft constraints. Most constraints are related to the government regulations, the preferences of the staff and the requirements of the hospitals. The objective of the research can be to minimize preference cost of resources [9, 16, 21], minimize the un-equilibrium of the schedule [21] and maximize satisfied demands [23]. Because in some models there are too many constraints [20, 27] and finding feasible solutions among all constraints may be impossible, the aim of these researches is to find feasible solutions satisfying all hard constraints while meeting as many soft constraints as possible.

For the proposed model, there are multiple sites. Costs associated with assigning resources to each site are different. Also, there are many periods. In each period, there are many tasks requiring joining of resources for operation. The resource composes of into two types: type1 and type2. Resources can be assigned to only one task in a period, and each task provides a different benefit. Resources can be assigned to only one site, and the objective of the model is to maximize the profit calculated from the benefit and the operation cost. The assignment model of Swangnop and Chaovallitwongse [28], which is a multi-period multi-site assignment problem with joint requirement of multiple resource types, can be adapted and applied to this problem. A mathematical model can be written as follows:

Index

- i = index for tasks; $i \in \{1, 2, 3, \dots, I\}$
 j = index for resources; $j \in \{1, 2, 3, \dots, J\}$
 p = index for periods; $p \in \{1, 2, 3, \dots, P\}$
 s = index for sites; $s \in \{1, 2, 3, \dots, S\}$

Set

- I_{ps} = set of task i occurring in site s in period p .

Parameters

- g_{jpi}^1 = 1 if the resource j in type 1 is qualified to do task i in period p .
 = 200 otherwise. [Big M value]
 g_{jpi}^2 = 1 if the resource j in type 2 is qualified to do task i in period p .
 = 200 otherwise. [Big M value]
 b_{pi}^1 = 1 if task i in period p requires resource type 1.
 = 0 otherwise.
 b_{pi}^2 = 1 if task i in period p requires resource type 2.
 = 0 otherwise.
 B_{pi} = benefit when task i in period p is executed.
 C_{js}^1 = operation cost when resource j in type 1 is assigned to site s .
 C_{js}^2 = operation cost when resource j in type 2 is assigned to site s .

Decision variables

- Y_{jpi}^1 = 1 if resource j in type 1 is assigned to task i in period p .
 = 0 otherwise.
 Y_{jpi}^2 = 1 if resource j in type 2 is assigned to task i in period p .

- Z_{js}^1 = 0 otherwise.
 Z_{js}^1 = 1 if resource j in type 1 is assigned to site s .
= 0 otherwise.
 Z_{js}^2 = 1 if resource j in type 2 is assigned to site s .
= 0 otherwise.
 W_{pi} = 1 if task i in period p is executed.
= 0 otherwise.

Objective function

$$\text{Maximize total profit} = \sum_{p=1}^P \sum_{i=1}^I B_{pi} W_{pi} - \sum_{r=1}^R \sum_{j=1}^J \sum_{s=1}^S C_{js}^1 Z_{js}^1 - \sum_{r=1}^R \sum_{j=1}^J \sum_{s=1}^S C_{js}^2 Z_{js}^2 \quad (1)$$

Constraints

$$\text{Qualification constraint [resource type 1]:} \quad \sum_{i=1}^I g_{jpi}^1 Y_{jpi}^1 \leq 1 ; \forall j \in \{1, \dots, J\}, p \in \{1, \dots, P\} \quad (2A)$$

$$\text{Qualification constraint [resource type 2]:} \quad \sum_{i=1}^I g_{jpi}^2 Y_{jpi}^2 \leq 1 ; \forall j \in \{1, \dots, J\}, p \in \{1, \dots, P\} \quad (2B)$$

$$\text{Location constraint [resource type 1]:} \quad \sum_{s=1}^S Z_{js}^1 = 1 ; \forall j \in \{1, \dots, J\} \quad (3A)$$

$$\text{Location constraint [resource type 2]:} \quad \sum_{s=1}^S Z_{js}^2 = 1 ; \forall j \in \{1, \dots, J\} \quad (3B)$$

$$\text{Joint requirement constraint [resource type 1]:} \quad \sum_{j=1}^J g_{jpi}^1 Y_{jpi}^1 = b_{pi}^1 W_{pi} ; \forall i \in \{1, \dots, I\}, p \in \{1, \dots, P\} \quad (4A)$$

$$\text{Joint requirement constraint [resource type 2]:} \quad \sum_{j=1}^J g_{jpi}^2 Y_{jpi}^2 = b_{pi}^2 W_{pi} ; \forall i \in \{1, \dots, I\}, p \in \{1, \dots, P\} \quad (4B)$$

$$\text{Available task constraint [resource type 1]:} \quad Z_{js}^1 \geq Y_{jpi}^1 ; \forall j \in \{1, \dots, J\}, p \in \{1, \dots, P\}, s \in \{1, \dots, S\}, i \in I_{ps} \quad (5A)$$

$$\text{Available task constraint [resource type 2]:} \quad Z_{js}^2 \geq Y_{jpi}^2 ; \forall j \in \{1, \dots, J\}, p \in \{1, \dots, P\}, s \in \{1, \dots, S\}, i \in I_{ps} \quad (5B)$$

The objective function, Eq. (1), maximizes the total profit, which is calculated from benefit and operation cost. Eqs. (2A) and (2B) enforce that only qualified resources can do tasks and each resource is assigned to only one task per period. Eqs. (3A) and (3B) enforce that each resource must be assigned to only one site. Eqs. (4A) and (4B) state that only qualified resources can do tasks and tasks can be done when all requirements of the resource are satisfied. Finally, Eqs. (5A) and (5B) indicate that resources can do only tasks in the site where they are assigned.

In the next section, a developed heuristic algorithm is described.

3. Heuristic Algorithm

There are many solution approaches for the multi-period multi-site assignment problem. Caunhye, Nie *et al.* [29] reviewed the problem of emergency logistics including solution approaches and techniques used in these models. Tzeng, Cheng *et al.* [15] developed a mathematical model for planning relief delivery and then transformed it to fuzzy multi-objective linear programming for making decisions. Fiedrich, Gehbauer *et al.* [26] applied simulated annealing and tabu search algorithm for allocating resources after earthquake disasters. Zhang, Li *et al.* [13] proposed a heuristic for emergency resource allocation problem concerning secondary disasters. Their algorithm is divided into three steps: finding an initial solution by linear relaxation, finding the solution for primary disaster by modifying the fractional parts in the initial solution and then applying local search techniques to assign resources to the secondary disaster points. For the

health staff scheduling problem, Trivedi and Warner [22] proposed modified a branch and bound algorithm for allocating float nurses. A problem of Costa Filho, Rivera Rocha *et al.* [12] concerning human resource allocation was modelled as a constraint satisfaction problem and used a backtracking search algorithm to find the solution. Others developed algorithms based on the metaheuristic concept, for example, ant colony optimization [11], genetic algorithm [16, 18, 21], memetic approach [20] and tabu search algorithm [19, 28]. The approach which is most related to the proposed problem is Tabu search algorithm of Swangnop and Chaovalitwongse [28]. However, there are many parameter settings in their algorithm. Fine tuning all parameters to suitable the proposed problem is not easy. Moreover, it takes a lot of computational time to obtain the solution and the quality of solutions tends to decrease when the problem size becomes large.

In this section, the proposed heuristic is presented. The algorithm is comprised of into two parts: generating an initial solution and improving solution. An initial solution from the first part is the feasible solution generated by CPLEX. The second part, improving solution, is the process of moving resources to new sites and assigning resources to tasks.

Part1: Generating an Initial Solution

This part aims to find a feasible initial solution. The process of this part can be written as follows.

1. Assign all resources to the site which has the lowest operation cost.
2. Assign resources to tasks by using CPLEX.

After the first step is done, all resources are assigned to sites. Then, the problem is decomposed into many sub-problems separated by site and period. Each of them is formulated as the assignment problem with one site and one period as shown in the following mathematical model.

Objective

$$\text{Maximize total profit} = \sum_{i=1}^I B_i W_i \quad (6)$$

Constraint

$$\text{Qualification constraint [resource type 1]:} \quad \sum_{i=1}^I g_{ji}^1 Y_{ji}^1 \leq 1 ; \forall j \in \{1, \dots, J\} \quad (7A)$$

$$\text{Qualification constraint [resource type 2]:} \quad \sum_{i=1}^I g_{ji}^2 Y_{ji}^2 \leq 1 ; \forall j \in \{1, \dots, J\} \quad (7B)$$

$$\text{Joint requirement constraint [resource type 1]:} \quad \sum_{j=1}^J g_{ji}^1 Y_{ji}^1 = b_i^1 W_i ; \forall i \in \{1, \dots, I\} \quad (8A)$$

$$\text{Joint requirement constraint [resource type 2]:} \quad \sum_{j=1}^J g_{ji}^2 Y_{ji}^2 = b_i^2 W_i ; \forall i \in \{1, \dots, I\} \quad (8B)$$

The decision is reduced to only assign resources to tasks (Y_{ji}^1 , Y_{ji}^2 and W_i). Terms of objective function and constraints are also reduced as shown in the Eq. (6)–(8). Each sub-problem is calculated by CPLEX to find the initial solution.

Part2: Improving Solution

This process intends to move resources to new sites and assign resources to tasks for improving the solution. Because, from the first part, all resources are assigned to the site which has the lowest operation cost, and the cost function in objective function has only the operation cost, the total cost from the initial solution is minimized. However, the method in part1 does not concern the qualification of resources and the requirement of task in each site. Term of benefit from this solution may not be good because of mismatching between skills of resource and requirements of task. An algorithm for allocating resources to better sites and assigning resources to better tasks is developed. In each move, there is a trade-off between the increasing of operation cost from moving resource to new site and the gain of benefit from assigning resource to new tasks. The process of algorithm in this part can be described in Fig. 1.

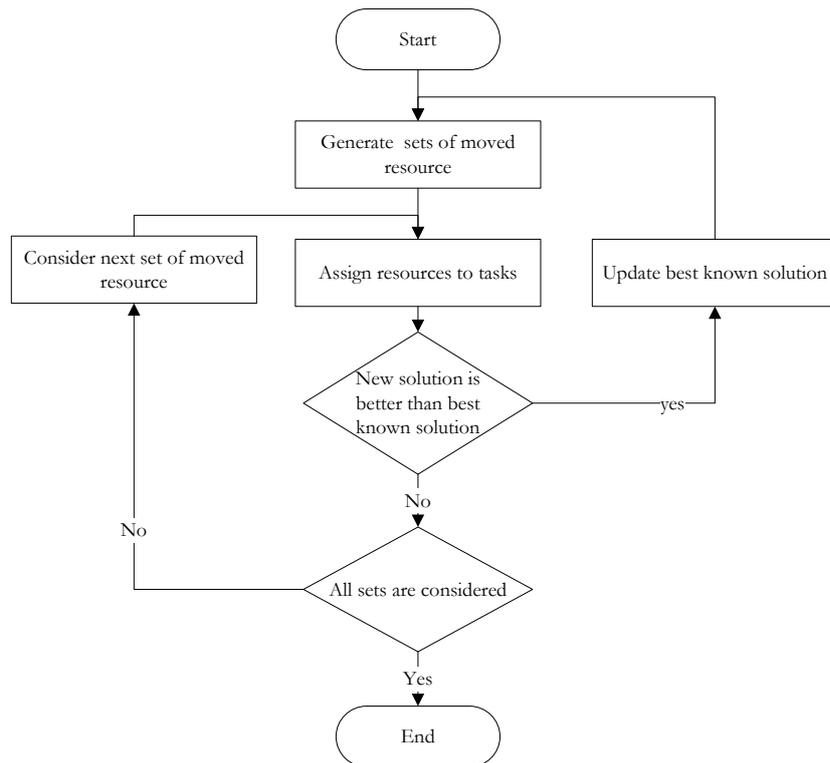


Fig. 1. Process of improving solution.

The algorithm starts from generating sets of moved resource, which consist of resources having potential to improve objective function when they are assigned to another sites. For each set, there is at most one resource per type. In a considered model, there are two types of resource so there are at most two resources in a set. There are many sets generated from this step; however, only one set of resources is chosen to be moved. After the resources are moved, the number of resources in some sites and periods are changed. The step of assigning resources to tasks in these sites and period is done. If the objective function of new solution is better than the best known solution, new solution is set to be the best known solution. Otherwise, the next set of resource is considered. The process of improving solution will be done iteratively until all sets of moved resource are considered and cannot improve objective function.

Main algorithms in this part are the algorithm for generating sets of moved resource and algorithm for assigning resources to tasks. The details of both algorithms are described below.

A. Algorithm for Generating Sets of Moved Resource

A set of moved resource consists of two elements: destination site and selected resources. The destination site is the site where resources are moved. The selected resources are the group of resource moved to the destination site. A structure of the set of moved resource can be written as follows: $\text{SetId}(m) = \{\text{Destination Site Id}(s) | \text{Selected resource Id}(j) \text{ from Type 1, Selected resource Id}(j') \text{ from Type 2}\}$, which means that in set m the resource j from type 1 and j' from type 2 are moved to site s .

Method to find destination site: All sites which have unassigned tasks (task that nobody does) are the destination sites.

Method to find selected resources: After the destination sites are defined, the selected resources moved to each destination site are chosen. The criteria for selection are the qualification of resources for doing unassigned tasks and the benefit and loss from moving resources. The process can be described as follows:

1. Consider one unassigned task in a destination site.
2. Categorize all resources qualified to do the unassigned task in step1 into four groups.

- Group1: Resource that is idle (resource that is not assigned to any task) and is in the destination site.
- Group2: Resource that is idle and is not in the destination site.
- Group3: Resource that is assigned to some tasks and is in the destination site.
- Group4: Resource that is assigned to some tasks and is not in the destination site.

All resources must be in only one group. There may be many resources in each group. Only one resource per group is chosen to be the selected resource. The criteria for selection are shown as follows.

- Group1: a selected resource is selected randomly.
- Group2: a selected resource is a resource that has the lowest changing site cost.

Changing site cost = $(C_{js}$ of resource in destination site) - $(C_{js}$ of resource in source site).

- Group3 and 4: a selected resource is a resource that has the lowest profit lost.

Profit lost = $(C_{js}$ of resource in destination site) - $(C_{js}$ of resource in source site) + (sum of benefits of all tasks that the resource does).

For unassigned tasks that require two resource types, this process is done separately by type.

After a destination site and selected resources in each group are known, the sets of moved resource are generated by selecting one resource in a group from each resource type to move to the destination site. All selected resources in all groups are considered for generating sets of moved resource. An example of this algorithm can be illustrated in Fig. 2.

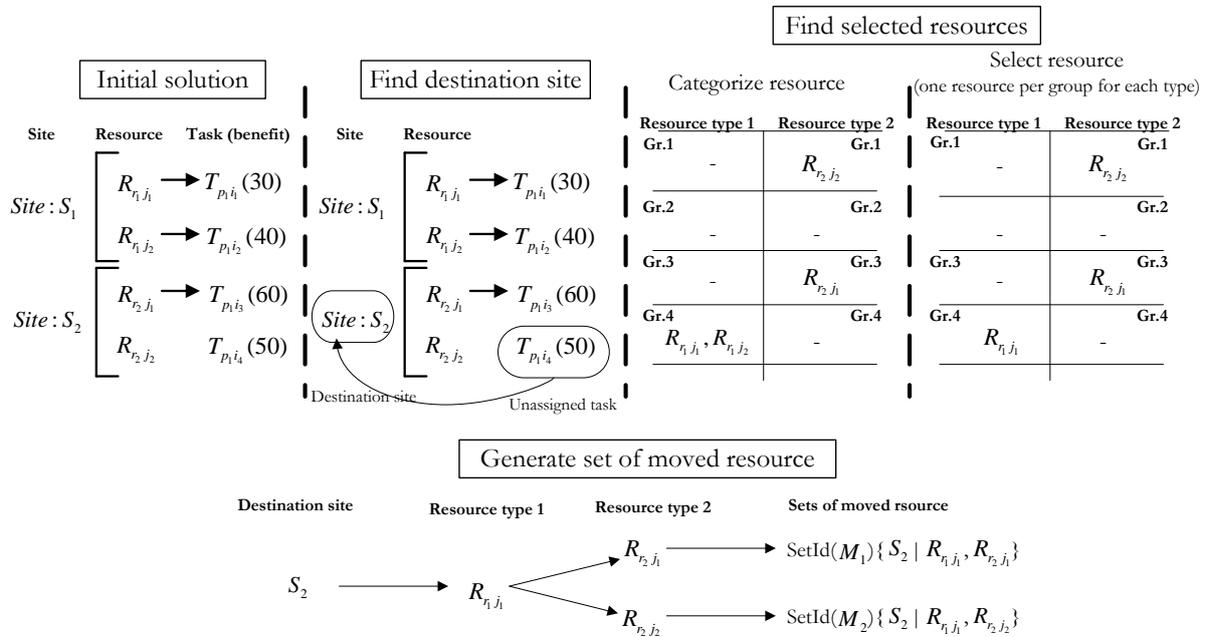


Fig. 2. Example of algorithm for generating sets of moved resource.

Assume there are two sites (S_1 and S_2), four resources (two resources for type 1 ($R_{r_1 j_1}$ and $R_{r_1 j_2}$) and two resources for type 2 ($R_{r_2 j_1}$ and $R_{r_2 j_2}$)), one period (p_1) and four task ($T_{p_1 i_1}$, $T_{p_1 i_2}$, $T_{p_1 i_3}$ and $T_{p_1 i_4}$). Task $T_{p_1 i_1}$ and $T_{p_1 i_2}$ are in site S_1 while task $T_{p_1 i_3}$ and $T_{p_1 i_4}$ are in site S_2 . Benefit of task $T_{p_1 i_1}$, $T_{p_1 i_2}$, $T_{p_1 i_3}$ and $T_{p_1 i_4}$ are 30, 40, 60 and 50 respectively. Task $T_{p_1 i_1}$, $T_{p_1 i_2}$ and $T_{p_1 i_3}$ are done by resource $R_{r_1 j_1}$, $R_{r_1 j_2}$ and $R_{r_2 j_1}$ while task $T_{p_1 i_4}$ are not assigned to any resource (unassigned task). The first step is to find the destination site, which is the site having unassigned tasks. Task $T_{p_1 i_4}$ is the unassigned task and is in site S_2 so the destination site is S_2 .

Then, task $T_{p_1 i_4}$, which is an unassigned task, is considered for finding the selected resources. Assume this task requires resource types 1 and 2 and all resources can do this task. For resource type 1 (r_1),

resources $R_{r_1 j_1}$ and $R_{r_1 j_2}$ are categorized into group 4. However, only one resource per group can be selected. The criterion for selection is the value of profit lost considering from the operation cost of resource and benefit from tasks which resources do. Assume the operation costs $C_{j_2 s_1}^{r_1}$ and $C_{j_2 s_2}^{r_1}$ of resource $R_{r_1 j_1}$ are 15 and 20 while $C_{j_2 s_1}^{r_1}$ and $C_{j_2 s_2}^{r_1}$ of resource $R_{r_1 j_2}$ are 15 and 25. The profit lost for resource $R_{r_1 j_1}$ and $R_{r_1 j_2}$ are $[(20-15) + 30] = 35$ and $[(25-15) + 40] = 50$ respectively. As a result, the resource $R_{r_1 j_1}$ is selected. Then, the resource type 2 (r_2) is considered. Resource $R_{r_2 j_1}$ is categorized into group 3 while resource $R_{r_2 j_2}$ is in group 1. There is one resource in groups 1 and 3 so both resources $R_{r_2 j_1}$ and $R_{r_2 j_2}$ are selected. In summary, there are three selected resources: one resource from type 1 ($R_{r_1 j_1}$) and two resources from type 2 ($R_{r_2 j_1}$ and $R_{r_2 j_2}$) and the destination site is site S_2 . Finally, sets of moved resource can be generated as follows: $\text{SetId}(M_1) \{ S_2 \mid R_{r_1 j_1}, R_{r_2 j_1} \}$ and $\text{SetId}(M_2) \{ S_2 \mid R_{r_1 j_1}, R_{r_2 j_2} \}$.

B. Algorithm for assigning resources to tasks

After the resources are moved, the number of resources in some sites and periods are changed. The destination site has more resources while the source site has less resources. This step is to assign resources to tasks in these sites and periods.

Focusing on the solution after moving resources to the destination site, there are one or two resources added to destination site and in source site there are one or two resources available because of cancelling some tasks as the example in Fig. 3.

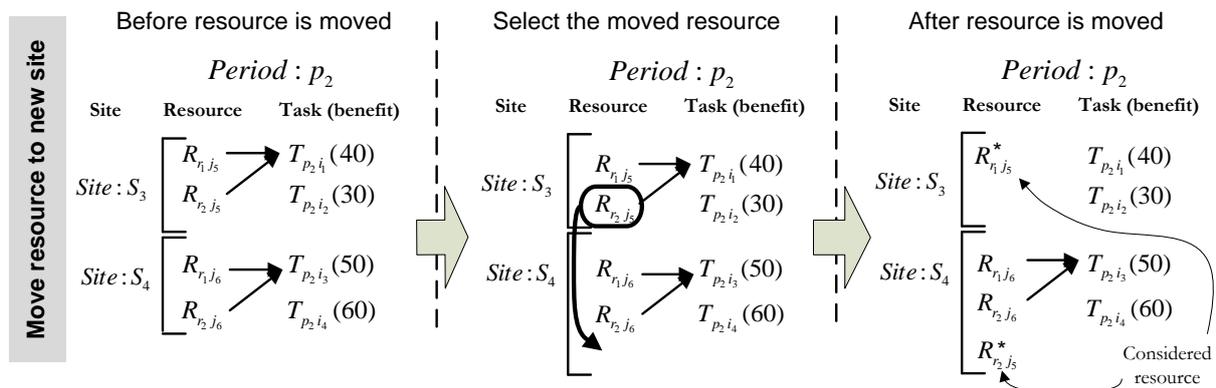


Fig. 3. Example of situation after the resource is moved.

In Fig. 3, period p_2 and site S_3 and S_4 are considered. Assume that resource $R_{r_2 j_5}$ is selected to be moved from site S_3 to site S_4 . After the resource is moved, task $T_{p_2 i_1}$ is cancelled or in other words, $T_{p_2 i_1}$ is changed from assigned task ($W_{p_2 i_1} = 1$) to unassigned task ($W_{p_2 i_1} = 0$) and in site S_3 resource $R_{r_1 j_5}$ is available. Now the resources whose status are changed are $R_{r_1 j_5}$ and $R_{r_2 j_5}$. This algorithm focuses on assigning these resources, named considered resources, to tasks.

Greedy search algorithm is applied to this step (assigning considered resources to task). The unassigned task which has the highest benefit and the considered resource can do is chosen to be assigned to these resources. For example, in site S_3 , in Fig. 4, assume resource $R_{r_1 j_5}$ can do task $T_{p_2 i_2}$. The highest benefit task that resource $R_{r_1 j_5}$ can do except task $T_{p_2 i_1}$ is task $T_{p_2 i_2}$ so resource $R_{r_1 j_5}$ is assigned to task $T_{p_2 i_2}$. If the highest benefit task cannot improve the objective function, the next highest benefit task is considered.

If the unassigned task requires two resource types, there may be one resource whose status is changed and available after assigning resource to the task. This available resource is defined to be the new

considered resource. For example, in site S_4 in Fig. 4, the considered resource is $R_{r_2j_5}$ (moved resource). From the concept of greedy search algorithm, task $T_{p_2i_4}$ which is unassigned task having highest benefit is considered. Assume task $T_{p_2i_4}$ requires two resource types and resource $R_{r_1j_6}$ and $R_{r_2j_5}$ can do this task. Resource $R_{r_1j_6}$ and $R_{r_2j_5}$ are assigned to task $T_{p_2i_4}$ and resource $R_{r_2j_5}$ is available because of cancelling task $T_{p_2i_3}$ and is defined to be the new considered resource. This process will be done iteratively until there is no considered resource or all tasks cannot improve the objective function.

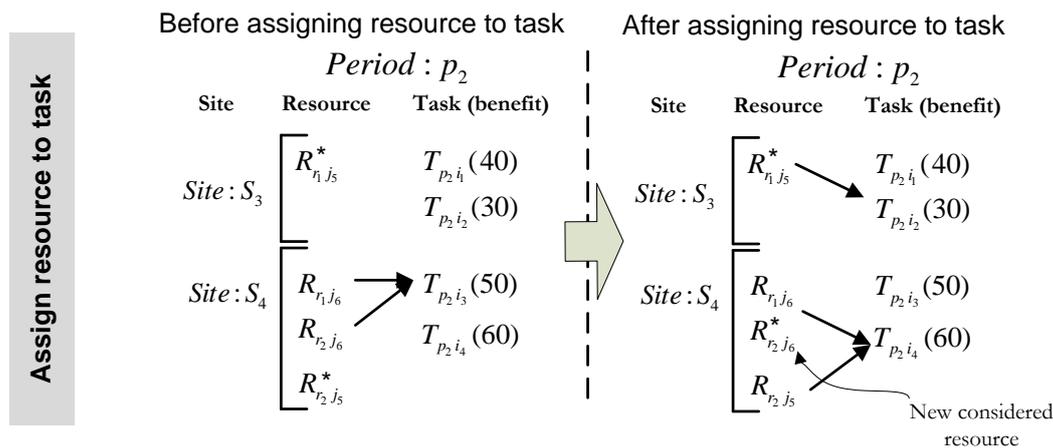


Fig. 4. Example of situation after the resource is assigned to task.

4. Computational Experiment

Because in this research the main consideration is joint requirement of two multi-skill resource types in the multi-period multi-site assignment problem, the objective of the experiment is to study the characteristic of the proposed problem when joint requirement is added and also evaluate the efficiency of the developed algorithm. All parameters in the experiment are set as shown in Table 1. The first column is the name of problem set and the rest columns are the parameter setting. There are eight problem sets: ProbA1-ProbA8. Each problem set, the number of resource is varied from 5 to 40. The number of period is fixed to 8 and 10 and the number of site is set to 5. Because tasks in the model can require one or two resource types for operation, to identify tasks in the experiment, the number of task and the ratio of task that requires one resource type and two resource types are specified. In this experiment, the number of task is set to 40 and 60 and the ratio is set to 0.50:0.50 and 0.25:0.75. The ratio 0.25:0.75 means that the number of task requiring one resource type is set to 25% of all tasks while the rest tasks (75% of all tasks) require two resource types. The algorithm is coded in C# 2010 and runs on a Windows 7 Ultimate with Intel Core i5-2410M, CPU 2.30GHz and RAM 4GB. Solutions from heuristic are compared with optimal solutions from a commercial optimization tool (ILOG CPLEX 12.6). For all problem sets, the operation cost and benefit are randomized uniformly between 2,000 to 10,000 and 400 to 4,000 respectively. The ratio of resource that can do each task is set to 0.4. For each problem, 5 tests are generated.

The results of the experiment are illustrated in Fig. 5 and Fig. 6. Figure 5 shows the computational time and optimal gap of ProbA1 to ProbA4 while Figure 6 shows the result of ProbA5 to ProbA8. In each problem set, the computational time and optimal gap of problems when the number of resource is set to 5, 10, 15, 20, 25, 30, 35 and 40 are plotted. An optimal gap in the experiment is calculated from $[(\text{solution of CPLEX}) - (\text{solution of heuristic})] * 100 / (\text{solution of CPLEX})$.

Table 1. Parameters of the first experiment.

Problem set	Number of resource	Number of period	Number of site	Number of task	Ratio of task requiring 1 resource type*	Ratio of task requiring 2 resource types*
ProbA1	5 - 40	8	5	40	0.50 (= 20 tasks)	0.50 (= 20 tasks)
ProbA2	5 - 40	8	5	60	0.50 (= 30 tasks)	0.50 (= 30 tasks)
ProbA3	5 - 40	10	5	40	0.50 (= 20 tasks)	0.50 (= 20 tasks)
ProbA4	5 - 40	10	5	60	0.50 (= 30 tasks)	0.50 (= 30 tasks)
ProbA5	5 - 40	8	5	40	0.25 (= 10 tasks)	0.75 (= 30 tasks)
ProbA6	5 - 40	8	5	60	0.25 (= 15 tasks)	0.75 (= 45 tasks)
ProbA7	5 - 40	10	5	40	0.25 (= 10 tasks)	0.75 (= 30 tasks)
ProbA8	5 - 40	10	5	60	0.25 (= 15 tasks)	0.75 (= 45 tasks)

*Tasks requiring one and two resource types are assigned randomly to all sites.

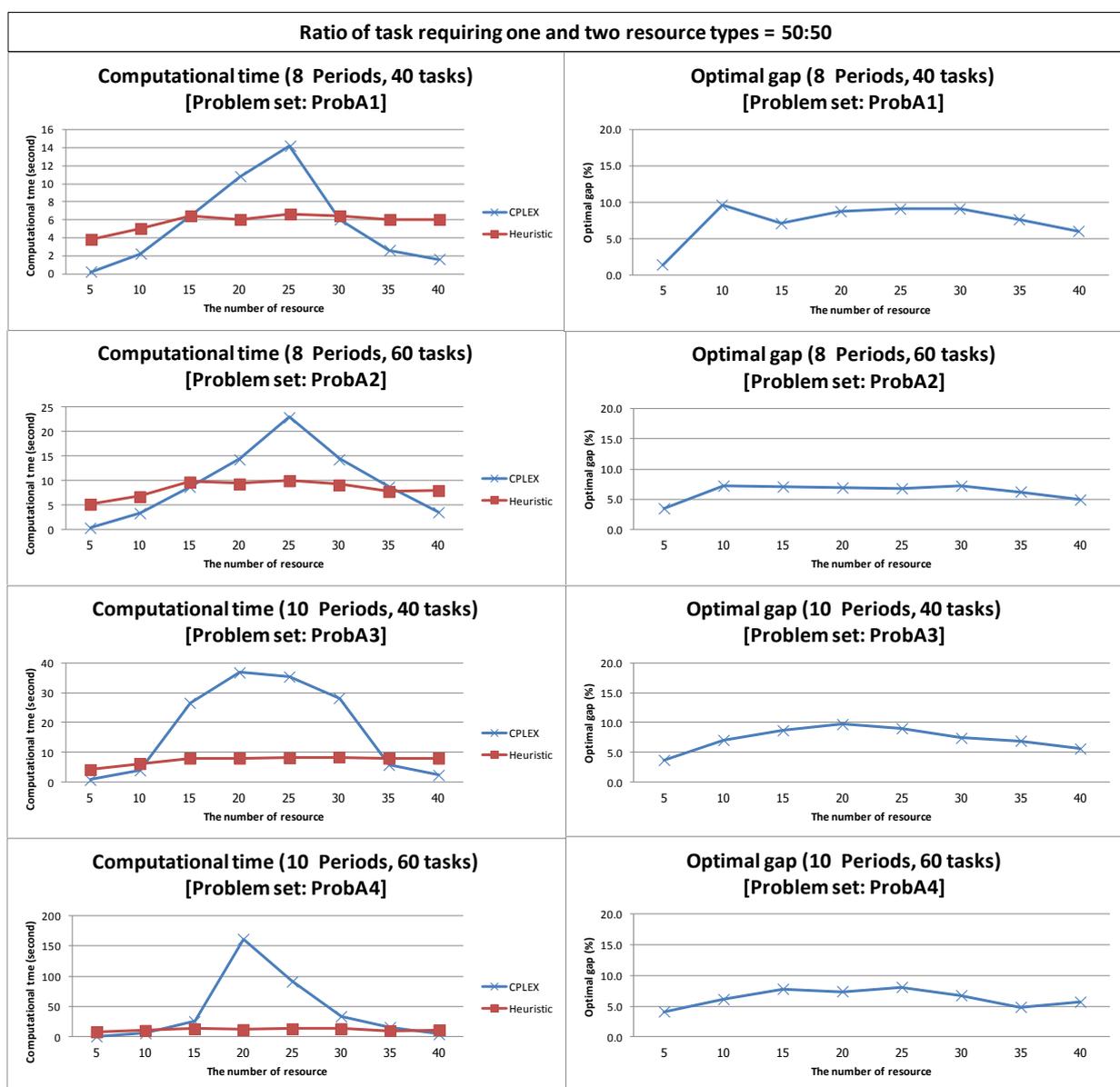


Fig. 5. Computational time and optimal gap of the first experiment [Ratio 50:50].

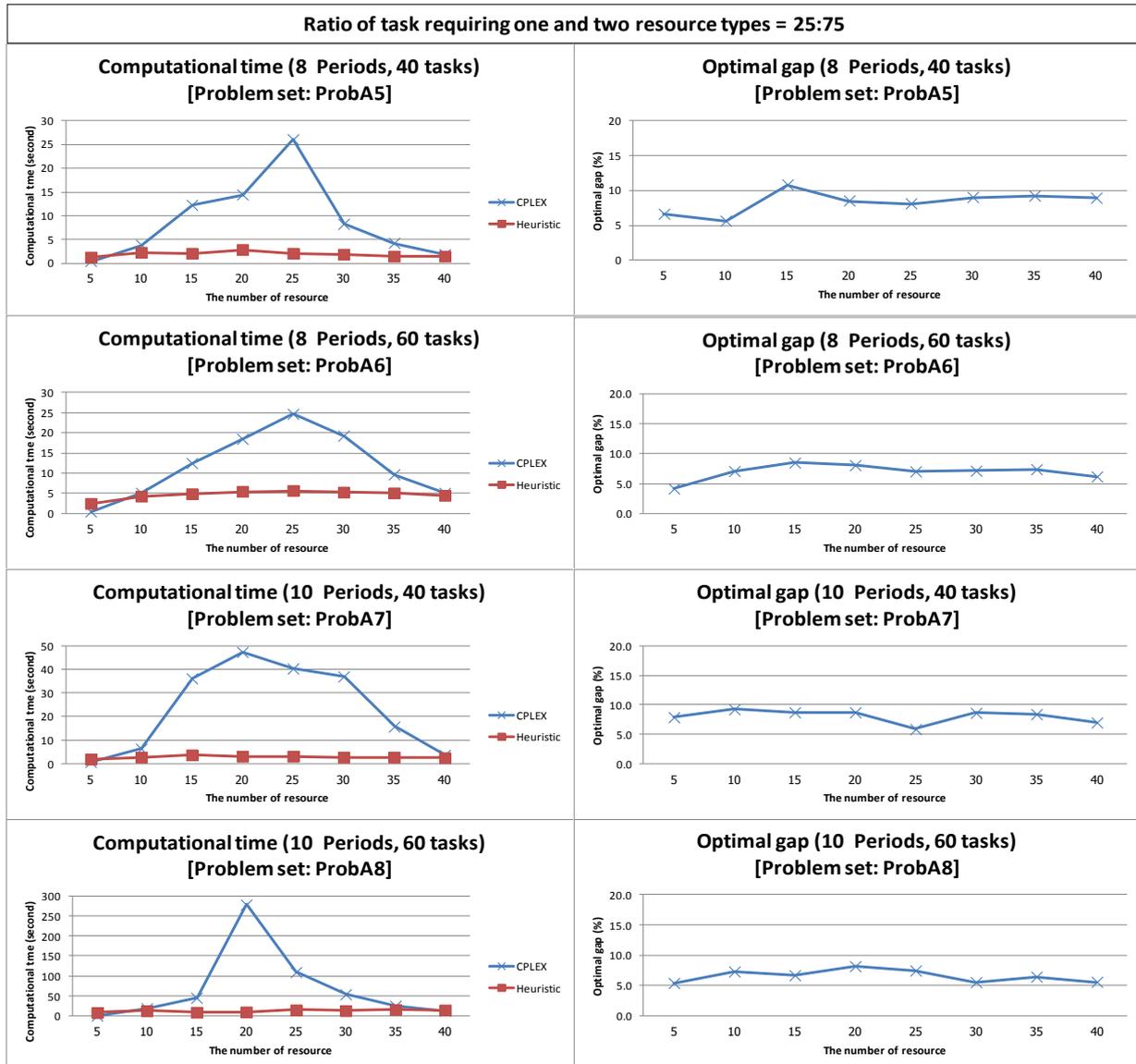


Fig. 6. Computational time and optimal gap of the first experiment [Ratio 25:75].

The finding from the results in Fig. 5 and Fig. 6 can be described in two aspects: the complexity of the problem and the efficiency of the developed algorithm.

The result shows that, from Fig. 5 and Fig. 6, the complexity of the problem increases when the number of period or task increases but the complexity does not always increase when the number of resource increases. Considering when all parameters are fixed and the number of resource is varied, such as in ProbA4 in Fig.5, the computational time of CPLEX is dropped when the number of resource is increased to some value. This is because when the number of resource is too large, it is easy to find resources for doing tasks. This result conforms to the generalized assignment problem (GAP) presented by Daz and Fernandez [30] whose computational time of the problem highly depends on the tightness of assigning resources to tasks and whose complexity does not always increase when the number of resource increases. In conclusion, the first finding from this experiment is that in each problem set, there is only one range of the number of resource that makes the problem most complex. For example, in the problem of 10 periods and 60 tasks in Fig.5 (ProbA4), the number of resources that makes the highest complexity is 20.

When focusing on the efficiency of the algorithm, the developed heuristic can find good solutions in all ranges of the number of resource in all problem sets (most optimal gaps are less than 10% and average optimal gap of all problem sets is 7.1%). Table 2 shows the computational time of the highest point of each problem set and minimum, maximum and average optimal gap of each problem set. The result from Table

2 shows that, at the highest point where the problem is the most complex, the developed heuristic can find good solutions in a short time comparing to CPLEX.

Table 2. The results from the first experiment.

Problem set	Computational time at the highest point (second)		Minimum optimal gap (%)	Maximum optimal gap(%)	Average optimal gap(%)
	CPLEX Time	Heuristic Time			
ProbA1	14	7	5.6	9.4	7.4
ProbA2	23	10	4.1	8.8	6.2
ProbA3	37	8	5.2	10.3	7.3
ProbA4	161	12	4.5	7.9	6.3
ProbA5	26	2	5.6	11.6	8.3
ProbA6	25	6	4.3	9.9	6.9
ProbA7	47	3	6.0	11.9	8.1
ProbA8	278	10	4.8	8.0	6.5

Moreover, in the proposed problem, all tasks do not require two resource types. The ratio of task requiring one and two resource types can be varied from 0 to 1. If ratio is set to 1:0, it means that all tasks require one resource type. Then, this problem is just a general multi-period multi-site assignment problem whose complexity should be less than the problem having tasks requiring joining of two resource types. On the other hand, if the ratio is set to 0:1, it means that all tasks require two resource types and the complexity of the problems should be high. This is confirmed by the result of the experiment. From this experiment, when comparing problems in the same size with different ratio, such as in ProbA4 of Fig. 5 and ProbA8 of Fig. 6, the computational time of CPLEX of problems in Fig. 6 whose ratio is set to 0.25:0.75 is longer than the computational time of problems in Fig. 5 whose ratio is set to 0.50:0.50.

To study more in the complexity of problems when the ratio is varied, the second experiment is conducted. In this experiment, the ratio of task requiring one and two resource type is varied from 1:0, 0.75:0.25, 0.5:0.5, 0.25:0.75 and 0:1. Other parameters are set as shown in Table 3. In this experiment, the problem is separated into two groups and five problems per group are generated: ProbB1.1 to ProbB1.5 for the first group and ProbB2.1 to ProbB2.5 for the second group. For the first group the problem is not complex and CPLEX can find optimal solutions in 50 seconds (when ratio is set to 0.25:0.75, referred from the first experiment), while for the second group the problem is more complex and CPLEX takes more than 350 seconds to find optimal solutions. The number of period, site and task are fixed to 10, 5 and 60, respectively. The number of resources is set to 15 for the first group and 20 for the second group.

Table3. Parameters of the second experiment.

Problem set	Number of resource	Number of period	Number of site	Number of task	Ratio of task requiring 1 resource type	Ratio of task requiring 2 resource types
ProbB1.1	15	10	5	60	1 (=60 tasks)	0 (=0 task)
ProbB1.2	15	10	5	60	0.75 (=45 tasks)	0.25 (=15 tasks)
ProbB1.3	15	10	5	60	0.50 (=30 tasks)	0.50 (=30 tasks)
ProbB1.4	15	10	5	60	0.25 (=15 tasks)	0.75 (=45 tasks)
ProbB1.5	15	10	5	60	0 (=0 task)	1 (=60 tasks)
ProbB2.1	20	10	5	60	1 (=60 tasks)	0 (=0 task)
ProbB2.2	20	10	5	60	0.75 (=45 tasks)	0.25 (=15 tasks)
ProbB2.3	20	10	5	60	0.50 (=30 tasks)	0.50 (=30 tasks)
ProbB2.4	20	10	5	60	0.25 (=15 tasks)	0.75 (=45 tasks)
ProbB2.5	20	10	5	60	0 (=0 task)	1 (=60 tasks)

The results of the experiment are shown in Table 4 and the computational time and optimal gap are illustrated in Fig. 7. The results show that the computational time of CPLEX dramatically increases when the ratio of task requiring two resource types increases. In contrast, the computational time of developed heuristic and the optimal gap slightly increase when the ratio increases. The quality of the solution is good in all problem sets (optimal gaps of all problems are less than 10% and average optimal gap of all problem sets is 7.4%).

Table 4. The results from the second experiment.

Problem set	CPLEX Time (sec)	Heuristic Time (sec)	Minimum optimal gap (%)	Maximum optimal gap (%)	Average optimal gap (%)
ProbB1.1	4	12	3.4	7.2	5.3
ProbB1.2	19	10	5.3	8.9	7.4
ProbB1.3	25	13	4.1	9.6	7.8
ProbB1.4	45	9	6.5	9.8	6.7
ProbB1.5	101	19	6.7	10.2	7.9
ProbB2.1	31	11	4.8	8.4	6.9
ProbB2.2	63	10	6.9	8.2	7.7
ProbB2.3	161	12	5.9	8.3	7.4
ProbB2.4	278	10	6.8	10.1	8.1
ProbB2.5	695	22	7.5	9.0	8.6

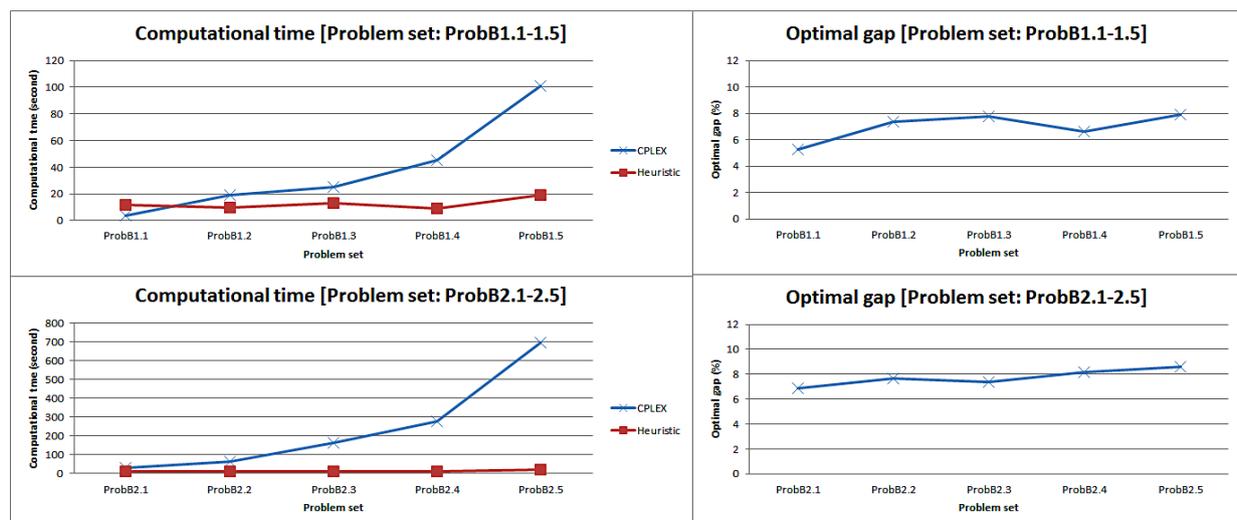


Fig. 7. Computational time and optimal gap of the second experiment.

In conclusion, the experiment shows that the computational time of CPLEX dramatically increases when the ratio of task requiring two resource types increases. When other parameters are fixed except the number of resources, there is only one range of the number of resources that makes the problem complex. The developed heuristic can find good solutions in a short time in all problem sets (average optimal gaps of the first and second experiment are 7.1% and 7.4%). The optimal gap slightly increases when the ratio of task requiring two resource types increases.

5. Conclusion

The purpose of this research is to develop a mathematical model and heuristic method for the multi-period multi-site assignment problem concerning joint requirement of two multi-skill resource types. This model can be seen in health resource planning in clinic networks. The developed heuristic is separated into two parts: finding an initial solution and improving the solution. The computational experiment is done for studying the characteristic of the proposed problem and evaluating the efficiency of the developed

algorithm. The results from experiments show that the joint requirement of resources drastically affects the complexity of the problem and the developed heuristic can find good solutions in a short time (average optimal gap of all test problems is 7.25%).

In the experiment of this research, we focus on studying the effect of parameters related to joint requirement of resources, so the further study is to study all parameters in the model. In addition, although the algorithm can provide good solutions, the optimal gap of some problems are rather high and tends to increase when the ratio of task requiring two resource types increases. The future study is also to improve the quality of the solution. Moreover, in actual operation of some problems, their resources can be rotated. Another subject of further study is to develop the assignment model concerning relocation of resources.

References

- [1] D. W. Pentico, "Assignment problems: A golden anniversary survey," *European Journal of Operational Research*, vol. 176, no. 2, pp. 774–793, 2007.
- [2] B. Gavish and H. Pirkul, "Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality," *Mathematical Programming*, vol. 31, no. 1, pp. 78–105, 1985.
- [3] J. B. Mazzola and A. W. Neebe, "Resource-constrained assignment scheduling," *Operations Research*, vol. 34, no. 4, pp. 560–572, 1986.
- [4] G. Caron, P. Hansen, and B. Jaumard, "The assignment problem with seniority and job priority constraints," *Operations Research*, vol. 47, no. 3, pp. 449–453, 1999.
- [5] B. Alidaee, H. Gao, and H. Wang, "A note on task assignment of several problems," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 1015–1018, 2010.
- [6] C. Rainwater, J. Geunes, and H. Edwin Romeijn, "The generalized assignment problem with flexible jobs," *Discrete Applied Mathematics*, vol. 157, no. 1, pp. 49–67, 2009.
- [7] A. Ravindran and V. Ramaswami, "On the bottleneck assignment problem," *Journal of Optimization Theory and Applications*, vol. 21, no. 4, pp. 451–458, 1977.
- [8] K. C. Gilbert and R. B. Hofstra, *Multidimensional Assignment Problems*. Blackwell Publishing Ltd, 1988, pp. 306–321.
- [9] L. S. Franz and J. L. Miller, "Scheduling medical residents to rotations: Solving the large-scale multiperiod staff assignment problem," *Operations Research*, vol. 41, no. 2, pp. 269–279, 1993.
- [10] X. Zhang and J. F. Bard, "A multi-period machine assignment problem," *European Journal of Operational Research*, vol. 170, no. 2, pp. 398–415, 2006.
- [11] W. J. Gutjahr and M. S. Rauner, "An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria," *Computers & Operations Research*, vol. 34, no. 3, pp. 642–666, 2007.
- [12] C. F. F. Costa Filho, D. A. Rivera Rocha, M. G. Fernandes Costa, and W. C. de Albuquerque Pereira, "Using Constraint Satisfaction Problem approach to solve human resource allocation problems in cooperative health services," *Expert Systems with Applications*, vol. 39, no. 1, pp. 385–394, 2012.
- [13] J.-H. Zhang, J. Li, and Z.-P. Liu, "Multiple-resource and multiple-depot emergency response problem considering secondary disasters," *Expert Systems with Applications*, vol. 39, no. 12, pp. 11066–11071, 2012.
- [14] L. Ozdamar, E. Ekinci, and B. Kucukyazici, "Emergency Logistics Planning in Natural Disasters," *Annals of Operations Research*, vol. 129, no. 1–4, pp. 217–245, 2004.
- [15] G.-H. Tzeng, H.-J. Cheng, and T. D. Huang, "Multi-objective optimal planning for designing relief delivery systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 6, pp. 673–686, 2007.
- [16] U. Aickelin and K. A. Dowsland, "An indirect Genetic Algorithm for a nurse-scheduling problem," *Computers & Operations Research*, vol. 31, no. 5, pp. 761–778, 2004.
- [17] M. W. Carter and S. D. Lapierre, "Scheduling emergency room physicians," *Health Care Management Science*, vol. 4, no. 4, pp. 347–360, 2001.
- [18] U. Aickelin and K. A. Dowsland, "Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem," *Journal of Scheduling*, vol. 31, pp. 139–153, 2000.
- [19] K. A. Dowsland, "Nurse scheduling with tabu search and strategic oscillation," *European Journal of Operational Research*, vol. 106, no. 2–3, pp. 393–407, 1998.
- [20] E. Burke, P. Cowling, P. D. Causmaecker, and G. V. Berghe, "A memetic approach to the nurse rostering problem," *Applied Intelligence*, vol. 15, no. 3, pp. 199–214, 2001.

- [21] C.-C. Tsai and S. H. A. Li, "A two-stage modeling with genetic algorithms for the nurse scheduling problem," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9506–9512, 2009.
- [22] V. M. Trivedi and D. M. Warner, "A branch and bound algorithm for optimum allocation of float nurses," *Management Science*, vol. 22, no. 9, pp. 972–981, 1976.
- [23] S. K. Goyal and J. P. Yadav, "Allocation of doctors to health centres in Haryana state of India—A case study," *The Journal of the Operational Research Society*, vol. 30, no. 5, pp. 427–431, 1979.
- [24] A. Volgenant, "A note on the assignment problem with seniority and job priority constraints," *European Journal of Operational Research*, vol. 154, no. 1, pp. 330–335, 2004.
- [25] J.-B. Sheu, "An emergency logistics distribution approach for quick response to urgent relief demand in disasters," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 6, pp. 687–709, 2007.
- [26] F. Fiedrich, F. Gehbauer, and U. Rickers, "Optimized resource allocation for emergency response after earthquake disasters," *Safety Science*, vol. 35, no. 1-3, pp. 41–57, 2000.
- [27] J. Puente, A. Gómez, I. Fernández, and P. Priore, "Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1232–1242, 2009.
- [28] S. Swangnop and P. Chaovalitwongse, "A two-step tabu search heuristic for multi-period multi-site assignment problem with joint requirement of multiple resource types," submitted for publication.
- [29] A. M. Caunhye, X. Nie, and S. Pokharel, "Optimization models in emergency logistics: A literature review," *Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 4–13, 2012.
- [30] J. A. Daz and E. Fernandez, "A tabu search heuristic for the generalized assignment problem," *European Journal of Operational Research*, vol. 132, no. 1, pp. 22–38, 2001.

