ปัญหาการมอบหมายงานที่มีหลายสถานีงานหลายช่วงเวลาโดยมีความต้องการใช้ทรัพยากรหลายประเภทร่วมกัน

นายสิรวิชญ์ สว่างนพ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556

MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM WITH JOINT REQUIREMENT OF MULTIPLE RESOURCE TYPES

Mr. Siravit Swangnop

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Industrial Engineering
Department of Industrial Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2013

Thesis Title	MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM WITH JOINT REQUIREMENT OF MULTIPLE RESOURCE TYPES
By	Mr. Siravit Swangnop
Field of Study	Industrial Engineering
Thesis Advisor	Assistant Professor Paveena Chaovalitwongse, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

....Dean of the Faculty of Engineering

(Professor Bundhit Eua-arporn, Ph.D.)

THESIS COMMITTEE

....Chairman

(Assistant Professor Manop Reodecha, Ph.D.)

....Thesis Advisor

(Assistant Professor Paveena Chaovalitwongse, Ph.D.)

....Examiner

(Assistant Professor Seeronk Prichanont, Ph.D.)

....Examiner

(Assistant Professor Wipawee Tharmmaphornphilas, Ph.D.)

....External Examiner

(Assistant Professor Rein Boondiskulchok, D.Eng.)

สิรวิชญ์ สว่างนพ : ปัญหาการมอบหมายงานที่มีหลายสถานีงานหลายช่วงเวลาโดยมีความต้องการใช้ทรัพยากรหลายประเภทร่วมกัน. (MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM WITH JOINT REQUIREMENT OF MULTIPLE RESOURCE TYPES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.ปวีณา เชาวลิตวงศ์, 102 หน้า.

ปัญหาการมอบหมายงานเป็นปัญหาที่มีการศึกษาและนำไปประยุกต์ใช้ในหลายอุตสาหกรรม วิทยานิพนธ์ฉบับนี้นำเสนอปัญหาการมอบหมายงานที่มีหลายช่วงเวลาหลายสถานีงาน โดยแต่ละงานต้องการใช้ทรัพยากรหลายประเภทร่วมกัน ในปัญหานี้ ทรัพยากรมีหลายประเภททักษะและงานต้องการใช้ทรัพยากรมากกว่าหนึ่งประเภทในการทำงาน การตัดสินใจในปัญหาดังกล่าว นอกจากจะต้องมอบหมายทรัพยากรให้ไปทำงานเหมือนปัญหาการมอบหมายงานทั่วไปแล้ว ยังต้องจัดสรรทรัพยากรให้ไปทำงานตามสถานีงานต่างๆในแต่ละช่วงเวลาอีกด้วย วิทยานิพนธ์ฉบับนี้ได้พัฒนาตัวแบบกำหนดการเชิงเส้นจำนวนเต็มและฮิวริสติกที่อาศัยหลักการของอัลกอลิทึมการค้นหาแบบทาบู (Two-step Tabu search heuristic) โดยที่กลยุทธ์การค้นหาคำตอบข้างเคียง (Neighborhood strategy) หน่วยความจำระยะสั้น (short-term memory) และหน่วยความจำระยะยาว (long-term memory) ได้ถูกออกแบบมาให้เหมาะสมกับปัญหาดังกล่าว นอกจากนี้ได้นำวัตถุประสงค์ทดแทน (Surrogate objective) มาใช้ประเมินคุณภาพของคำตอบข้างเคียงด้วย ในการเพิ่มความเร็วของการค้นหาคำตอบจะพิจารณาเฉพาะคำตอบข้างเคียงที่ดีเท่านั้น คุณภาพคำตอบจากฮิวริสติกที่พัฒนาขึ้นถูกประเมินโดยการนำไปเปรียบเทียบกับคำตอบที่ดีที่สุด (Optimum solution) ที่ได้จาก CPLEX โดยที่ผลการทดลองแสดงให้เห็นว่าสำหรับปัญหาขนาดเล็ก ฮิวริสติกที่นำเสนอสามารถหาคำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุดได้ โดยมีค่าเฉลี่ยช่วงห่างระหว่างคำตอบที่ได้จากฮิวริสติกและ CPLEX (Optimum gap) เท่ากับ 0.09% สำหรับปัญหาขนาดกลาง อัลกอริทึมสามารถหาคำตอบที่ดีได้ภายในเวลาที่เหมาะสม โดยมีค่าเฉลี่ยช่วงห่างระหว่างคำตอบเท่ากับ 4.42% และสำหรับปัญหาขนาดใหญ่ที่จำกัดเวลาการหาคำตอบของ CPLEX ไว้ที่ 10 ชั่วโมง ค่าเฉลี่ยช่วงห่างระหว่างคำตอบที่ได้จากฮิวริสติกและค่ามากสุดที่เป็นไปได้ของคำตอบ (Upper bound) จาก CPLEX เท่ากับ 8.28%

ภาควิชา วิศวกรรมอุตสาหการ ลายมือชื่อนิสิต ....

สาขาวิชา วิศวกรรมอุตสาหการ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก ....

ปีการศึกษา 2556

5171835921 : MAJOR INDUSTRIAL ENGINEERING
KEYWORDS: MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM/ JOINT REQUIREMENT/ RESOURCE TYPE/ HEURISTIC/ TABU SEARCH

SIRAVIT SWANGNOP: MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM WITH JOINT REQUIREMENT OF MULTIPLE RESOURCE TYPES. ADVISOR: ASST. PROF. PAVEENA CHAOVALITWONGSE, Ph.D., 102 pp.

An assignment problem has been extensively studied and applied in many industries. This research proposes a multi-period multi-site assignment problem with joint requirement of multiple resource types. In this problem, there are many multi-skill resource types, and each task requires joint of more than one resource type to operate. The decisions in the proposed model are not only assigning resources to tasks as in classic assignment problems but also allocating resources to sites in each period. An integer linear programming model and a heuristic approach based on Tabu search algorithm (Two–step Tabu search heuristic) are developed. The specified neighborhood strategy, short-term memory and long-term memory are designed for the addressed problem in order to generate an efficient move to improve solutions. In addition, the surrogate objective is introduced to evaluate the quality of neighborhoods, and only good neighborhoods are considered to increase search speed. The quality of solutions from the developed heuristic are compared with optimal solutions from CPLEX. For small size problems, the result shows that the proposed heuristic can find solutions close to optimum in most problems at the average optimal gap of 0.09%. For medium size problems, the algorithm can provide good solutions in a reasonable time at the average optimal gap of 4.42%. Finally, for large size problems whose computational time of CPLEX is limited to 10 hours, the average gap between solutions from heuristic and upper bounds from CPLEX is 8.28%.

Department:	Industrial Engineering	Student's Signature	....
Field of Study:	Industrial Engineering	Advisor's Signature	....
Academic Year:	2013		

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Assistant Professor Paveena Chaovalitwongse, for her guidance, advice and support during my student life. In addition, I would like to thank Assistant Professor Rein Boondiskulchok for guiding and supporting all my researches both in master and doctoral degree. Special thanks to Naragain Phumchusri and Oran Kittithreerapronchai for teaching me how to be a good speaker in both academic forum and general meeting and advising me how to improve my research. I would like to thank Anawat Ariyasujjakorn for teaching me how to code and develop algorithm by C#. Thanks to Sakol suethanapornkul for teaching me how to develop a good paper and give the guidance and feedback on my paper. Lastly, I would like to thank my family for all their support through my study and my life.

CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER I
INTRODUCTION

1.1 Introduction

An assignment problem has been extensively studied and applied in many industries, namely dairy [1], clothing [2], mining [3], airlines [4], automated manufacturing [5] and service industries [6]. First appearing in 1952 [7], the classic assignment problem is to find a one-to-one matching between n tasks and m agents and the objective function is to minimize the total cost. Over the past few decades, the classic assignment problem has been extended and many variations of the assignment problem are proposed, for example, variation in objective function such as maximizing profit[8, 9] or minimizing the maximum number of travelling time[10], variation in planning period such as three-dimensional assignment problem [11], multi-period assignment problems for medical residents [12], multi-period machine assignment [13] or variation in task and resource such as multi-resource generalized assignment problem [14], resource-constrained assignment scheduling [15], assignment problem with seniority and job priority constraints [16] and generalization of multi-resource generalized assignment problem[9].

A multi-period multi-site assignment problem is one extension of the classic assignment problem. The number of site and period 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 [17, 18]. 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 [19-21], and the problem of health staff scheduling, which is a problem of allocating or assigning physicians or nurses to shifts, wards and hospitals [17, 18, 22-29].

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 physician or nurse. Goyal and Yadav [22], Trivedi and Warner

[28], Gutjahr and Rauner [17], Aickelin and Dowsland [24], Dowsland [25], Burke, Cowling et al. [26] and Tsai and Li [27] developed nurse scheduling models whereas Carter and Lapierre [23], Costa Filho, Rivera Rocha et al.[18] and Goyal and Yadav [29] proposed physician 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.[30] , Ozdamar et al.[20] and Tzeng et al.[21] 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 many types such as physicians, nurses or medical equipments and their tasks or treatments require the joining of more than one 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 research, we are interested in developing multi-period multi-site assignment model concerning joint requirement of multiple resource types and also proposing the solution method for finding solutions.

1.2 Problem statement

As described in previous section, this research focused on developing multi-period multi-site assignment problem with joint requirement of multiple resource types.

Objective functions, decision variables and constraints of multi-period multi-site assignment problem in the previous studies are different depending on the related applications.

In the problem of emergency resource allocation, sites are emergency depots, public areas, hospitals or disaster sites; resources are medical supplies, equipments or staff. The decision on most models is to allocate resources to disaster sites in as short time 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 [19-21, 31]. Some models are also concerned with the transportation of resource or vehicle routing so they also include the number of vehicle required for transporting resources between nodes into the decision [20]. Planning horizon are always multiple periods [19-21, 31]. Conditions or constraints in most researches 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 [19, 21], minimize fatalities in the search and rescue period [32], minimize unsatisfied demands [20], minimize operation cost [21] and maximize fairness of resource distribution [21].

In the problem of health staff scheduling, sites are wards, departments or hospitals and resources are physicians 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[22-27], to wards in a hospital [28] or to hospitals in a considered area [17, 18, 29]. Planning horizon can be one shift in advance [28] or many time periods [17, 18, 22-27]. 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 [12, 22, 27], minimize the un-equilibrium of the schedule [27] and maximize satisfied demands [29]. Because in some models there are too many constraints [26, 33] 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.

In this research, the proposed multi-period multi-site assignment problem is motivated by the problem of health resource planning in clinic networks. In this problem, there are many service locations located in many regions and their health resources such as physician, nurses or medical equipments are seen as the pool of resource which can be assigned to any locations. The resources are divided into many types and the treatments or demands require joint of many resource types for operation such as joint of physician and nurse or joint of nurse and medical equipment. Each service location has many customers which require different treatments and each resource has specific skills which can treat only some customers. They must be rotated to many service locations to treat as many customers as possible but the cost occurring from rotating their resources to each service location must be concerned. The clinic network will support all transportation costs and accommodation costs (operation cost) of their resources. The planner has to decide where their resources should be assigned to fulfill demands in each day to maximize total profit (Complete details of the clinic network are described in Appendix A).

1.2.1 Problem description

From the characteristic of clinic network, the proposed model can be described as follows. Our resources are multi-skill resources which are divided into many resource types. Tasks in the problem can require one or more than one resource type for operations and only qualified resources can do tasks. Each task provides different benefits. There are many working day and in each day working time is divided into many time periods. In each time period, there are many tasks and resource can do only one task in a period. There are many working sites and resources can be rotated to any sites at the end of the day. After resources finish their tasks on the last date, they must be returned to sites where they are assigned on the first date. Operation costs of each resource and transportation costs between sites are different. The objective is to maximize total profit which is calculated from benefit, operation cost and transportation cost.

An example of the proposed model can be described through Fig.I-1.

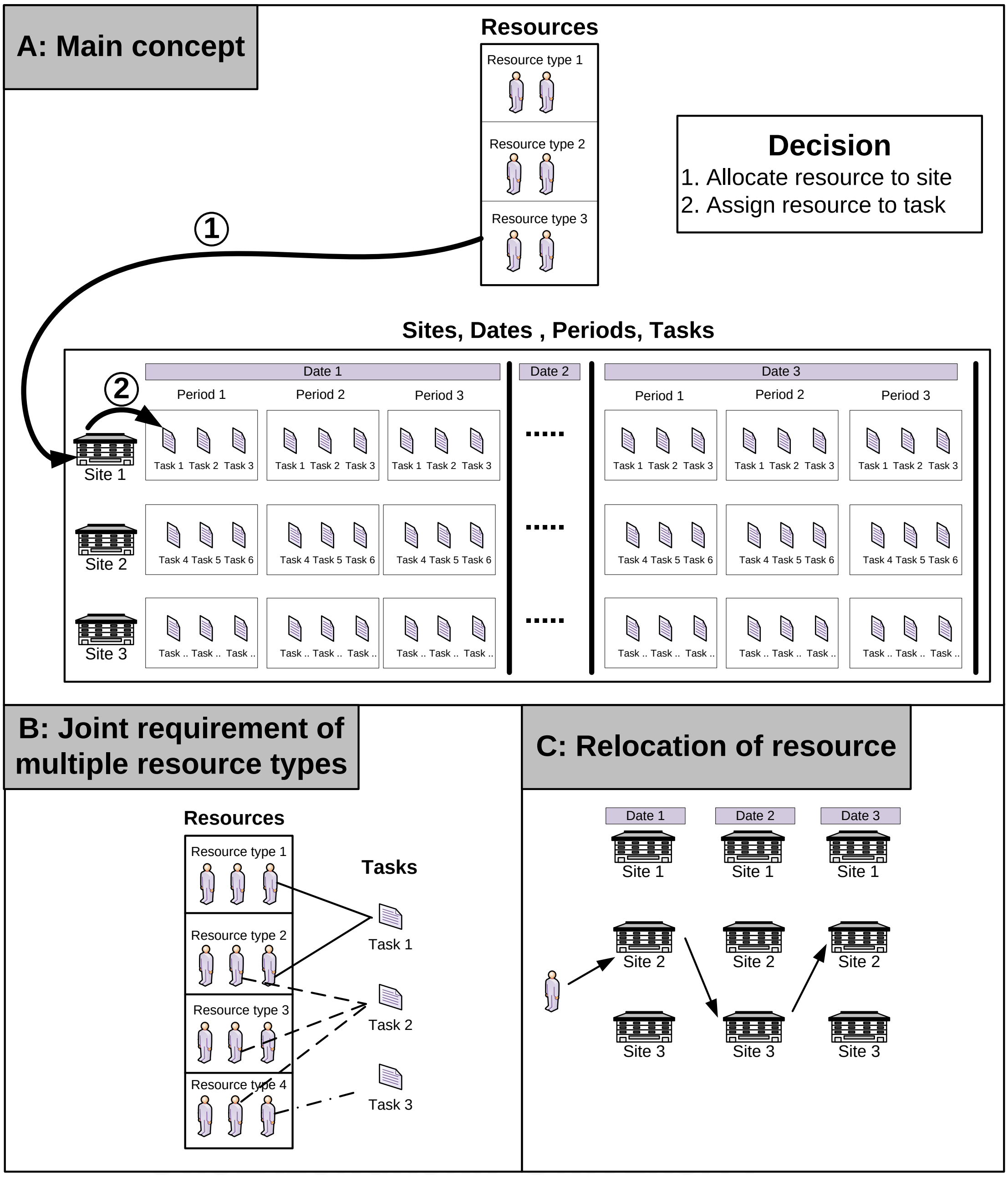


Figure I-1 An example of the proposed model

In Fig.I-1(A), there are three resource types (Resource type 1, Resource type 2 and Resource type 3) and two resources per type. There are three sites (Site 1, Site 2 and Site 3) and three days (Date 1, Date 2 and Date 3). In each day, there are three periods (Period 1, Period 2 and Period 3). In each period, there are many tasks. The resource can be assigned to only one site per day and done only one task per period.

In Fig.I-1(B), examples of joint requirement are shown. Task 1 requires Resource type 1 and Resource type 2 whereas Task 2 requires Resource type 1, 2 and 3. For Task 3, only Resource type 4 is required.

In Fig.I-1(C), an example of relocation of resource is illustrated. The resource in this example is assigned to Site 2 in Date1, Site 3 in Date 2 and Site 2 in Date 3.

The decision is to allocate resources to sites and assign resources to tasks to maximize total profit which is calculated from total benefit, total operation cost and total transportation cost. Total benefit is calculated from benefit from executed tasks in each period. Total operation cost is calculated from cost of assigning each resource to site and total transportation cost is calculated from cost of relocating resources to another site at the end of each day.

Generally, a multi-period assignment problem is in the class of NP problems [34]. In our model, the joint requirement of multiple resources in multiple sites is added. Finding optimal solutions is hard when the problems become large and impossible for real size problems. Consequently, in addition to develop mathematical model, heuristic algorithm is developed for finding the solution.

1.3 Research objective

An objective of this research is to develop mathematical model and heuristic for multi-period multi-site assignment problem with joint requirement of multiple resource types.

1.4 Research scope

- This research focuses on developing mathematical model and heuristic for multi-period multi-site assignment problem with joint requirement of multiple resource types.
- The developed heuristic is evaluated by comparing with optimal or best solutions found by CPLEX.
- All parameters, which are benefit, operation cost, transportation cost, resource, demand and joint requirement condition, are deterministic and given.

1.5 Research contribution

The contribution from this research can be described into two aspects: contribution in academic researches and contribution for real life application.

Contribution in academic researches

In this study, we extend the assignment problem by proposing the joint requirement of multiple resource types in the multi-period multi-site assignment problem. Most multi-period multi-site assignment models do not concern joint requirement of multiple resource types as in the proposed model. It can be said that we introduce a new problem to the series of assignment model.

Contribution for real life applications

Most researches related to health resource planning consider only one type of resource which is physician, nurse, or medical equipment. However, there is a case in which joint requirement of resources is required such as in clinic network business. The proposed model can be applied to this application and helps decision makers schedule their resources efficiently.

1.6 Research methodology

The research methodology can be described in Fig.I-2.

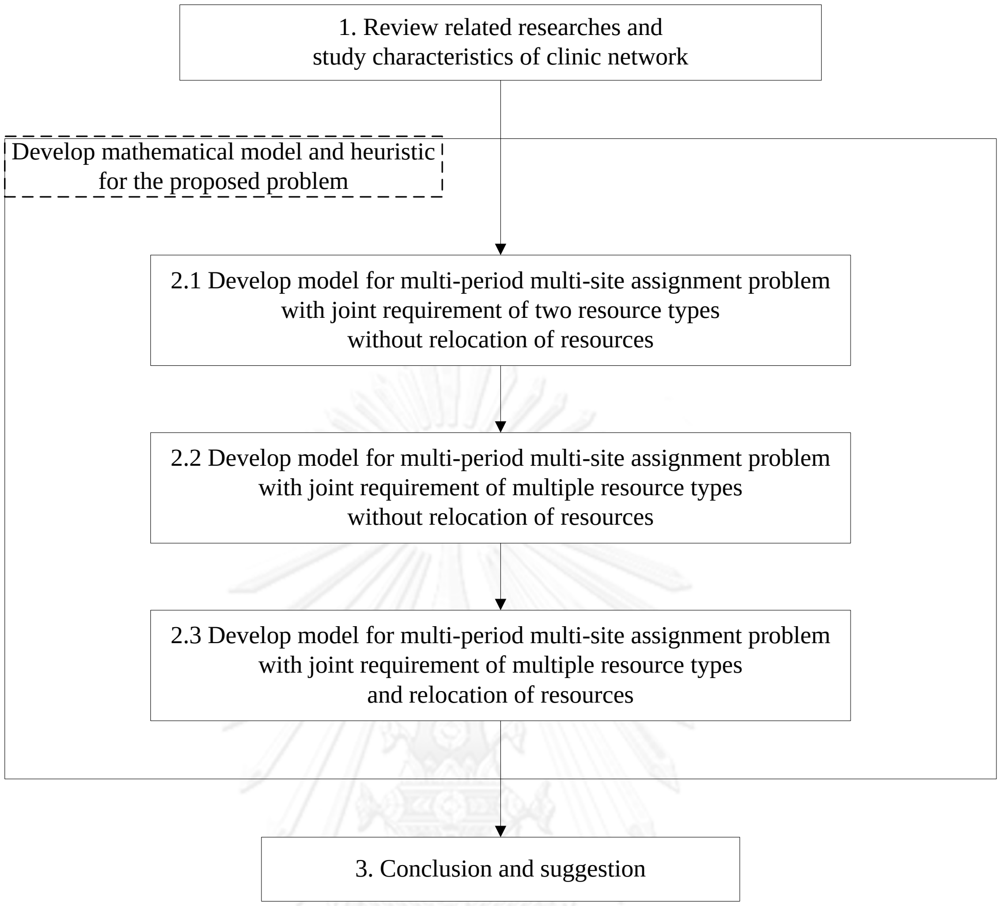


Figure I-2 Research methodology

The methodology can be divided into three steps. The first step is to review related researches and study characteristics of clinic network. Then, in the second step, the process of developing mathematical model and heuristic is done. This step is divided into three phases (step 2.1 – step 2.3). From the first phase to the last phase, the model is developed from simple assignment model (some dimensions of the proposed problem are reduced) to complex assignment model (complete proposed problem). In the first phase (step 2.1), the problem has only two resource types and relocation of resources is not allowed. In the next phase (step 2.2), the number of resource type is increased to more than two and in the last phase (step 2.3), the model which has multiple resource types and has relocation of resource is developed. In each phase the mathematical models and heuristics are developed. Finally, the conclusion and suggestion are discussed. Details of each step can be described as follows.

Step1: Review related problems and study characteristics of clinic network.

- Study the variations in assignment problem. [in Section 2.1]
- Study the modeling approaches and solution methods in multi-period multi-site assignment problem. [in Section 2.2]
- Study the characteristics of clinic network. [in Appendix A]

Step 2.1: Develop model for multi-period multi-site assignment problem with joint requirement of two resource types without relocation of resources (MM-J2-NoRe). [in Chapter 3]

- In this model, the number of resource type is fixed to 2 and resources are not allowed to be rotated.
- An objective of this step is to develop mathematical model and heuristic and also to study the characteristic of the multi-period multi-site assignment problem when the joint requirement of resource is added.

Step 2.2: Develop model for multi-period multi-site assignment problem with joint requirement of multiple resource types without relocation of resources (MM-JM-NoRe). [in Chapter 4]

- In this model, the number of resource type is not limited to 2. However, resources are still not allowed to be rotated.
- An objective of this step is to develop mathematical model and heuristic, which is based on the model and solution method in step 2.1.

Step 2.3: Develop model for multi-period multi-site assignment problem with joint requirement of multiple resource types and relocation of resources (MM-JM-Re). [in Chapter 5]

- In this model, the number of resource type is not limited to 2 and relocation of resources is allowed.
- An objective of this step is to develop mathematical model and heuristic for the complete proposed problem.

Step 3: Conclusion and suggestion

- Conclude all models and solution methods. [in Section 6.1 and 6.2]

- Suggest the further study. [in Section 6.3]

1.7 Dissertation structure

The structure of this research can be described as follows. The related researches are reviewed in Chapter II. In Chapter III, IV and V, the mathematical model and heuristic for each problem are proposed. Finally, the conclusion and suggestion for further study are discussed in Chapter VI.

CHAPTER II

LITERATURE REVIEW

In this chapter, the related researches are reviewed. Because the proposed model is extended from the multi-period multi-site assignment problem, the assignment problem and its extension are reviewed (Section 2.1). To study in detail of multi-period multi-site assignment problem, the related applications in multi-period multi-site assignment problem and solution approaches of this problem are also studied (Section 2.2). Finally, in the last section (Section 2.3) all reviews are summarized.

2.1 Assignment problem and its extensions

The name of assignment problem has first appeared in 1952 [7]. The classic assignment problem is to find a one-to-one matching between n tasks and m agents and the objective function is to minimize the total cost [7]. The mathematical model of the classic assignment problem can be given as:

Minimize $$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{2.1}$$

Subject to $$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,2,\ldots,n \tag{2.2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,\ldots,n \tag{2.3}$$

$$x_{ij} = 0 \text{ or } 1 \tag{2.4}$$

; where x_{ij} = 1 if agent i is assigned to task j, 0 if not, and c_{ij} = the cost of assigning agent i to task j.

Over the past many years, the classic assignment problem has been extended and many variations of the assignment problem have been proposed. The interesting variations are the variation in objective, variation in agent and variation in planning period.

- Variation in objective

The objective of the classic assignment problem, which is minimizing total cost, has modified. For example, objective function of the Ravindran and Ramaswami [10] is to minimize the maximum number of the time. In this research, there are warehouses and markets in the system. Planners have to determine how to transport perishable goods from warehouses to markets without spoilage. This problem is called the bottleneck assignment problem. Another example is the research of Rainwater, et al. [8]. The objective of the research is to maximize profit which consists of cost and benefit. When agents are assigned to the jobs, they can get the benefit. The objective of the Alidaee, et al. [9] is also to maximize the profit but the profit of this model consists only of the benefit. Therefore, they try to do as many jobs as they can to maximize total benefit. The application of this problem can be found in the maintenance department in which the decision of the model is to choose as many maintenance activities as possible to maximize total benefit.

- Variation in agent

The variation in agent in this meaning is the variation of the skill that agents have and the required skills for operating each task. The agent in classic assignment problem can be assigned to any tasks but in many situations, some job can be done by only qualified agents. Caron, et al. [16] proposed the assignment model that concerned the seniority of the task and priority of the job. Therefore, each agent has different seniorities and each job has different priorities.

Another and important meaning of the variation in agent is that one agent has many skills and they can do many tasks as long as they have enough capability. This problem is called the generalized assignment problem (GAP). This model assumes that each task will be assigned to one agent, while an agent may be assigned to more than one task.

Minimize $$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad (2.5)$$

$$\text{Subject to} \quad \sum_{j=1}^{n} a_{ij}x_{ij} \le b_i \qquad i=1,\ldots,m \qquad (2.6)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j=1,\ldots,n \qquad (2.7)$$

$$x_{ij} \in \{0,1\} \qquad i=1,\ldots,m;\ j=1,\ldots,n \qquad (2.8)$$

; where x_{ij} = 1 if agent i is assigned to task j, 0 if not, c_{ij} = the cost of assigning agent i to task j, a_{ij} is the amount of agent i's capacity used if that agent is assigned to task j, and b_{ij} is the available capacity of agent i.

Generalized assignment problem is the standard problem for any other researchers who would like to develop the multi-skill assignment problem or resource-constrained assignment problem. Some extensions of the GAP are shown as follow. Gavish and Pirkul [14] proposed the multi-resource generalized assignment problem (MRGAP), which extends GAP by allowing agents to have multiple resources and their resources are consumed when accomplishing their tasks. Mazzola and Neebe [15] defined the assignment problem with side constraints (APSC) which extends GAP as MRGAP but the resources in the system are not separated by individual. All agents can use the resources of the system until those resources are out. There are the pools of resources that everyone can use. Another example of APSC can be found in Foulds and Wilson [35]. This problem is to assign the tools to the slots in a tool carousel for a flexible manufacturing system. Toktas et al. [36] combined the characteristics of the GAP and MRGAP with the APSC and generated two additional problems: collectively capacitated generalized assignment problem (CCGAP) and assignment problem with individual capacities (AIPC). GAP and MRGAP is one-to-many assignment while APSC is one-to-one assignment. In addition, resources in GAP and MRGAP are separated by individual while resources in APSC are pooled in the center and everyone can use. The CCGAP is the combination of GAP and APSC and the APIC is the combination of the MRGAP and APSC. Alidaee, et al. [9] presented the assignment model that includes the model of MRGAP in the previous research papers as special cases.

This new model is called generalization of MRGAP (GMRGAP). This model has many resources and their resources have many skills. In addition, there are many tasks and their tasks have many operations. If tasks are chosen for an assignment, all its operations must be completed. When task are done, they will get the benefit. The objective of this model is to choose as many tasks as possible to finish the operation while maximizing the total expected benefit. Their resources are limited so they cannot do all tasks. This problem generalizes the MRGAP, which is NP-hard problem, so GMRGAP is NP-hard problem too. They do a computational experiment for testing the effectiveness of the model compared with the GAP. The result of this test shows that the number of variables, the optimality gap, and the average CPU time to reach the best solution of GMRGAP is less than the GAP. The characteristics of the GMRGAP close to the proposed model, which concerns limited multi-skill resources and joint of resources for doing tasks. However, the objective function of this model consists of benefit while the proposed model consists of benefit and cost. In addition, this model has only one period and one site but the proposed model has multiple periods and multiple sites.

- Variation in planning period

The number of period for the classic assignment problem is only one while the planning period of many models is multiple periods. The assignment model which considers many periods can be classified to the multi-dimension assignment problem or multi-period assignment problem. The multi-dimension problem is the matching the member of three or more sets such as assigning students and teachers to classes and time slots. Gilbert and Hofstra [11] classified two different versions of the three-dimensional problem: planar and axial. The example of planar three-dimensional problem is the scheduling of meetings between venders and customers over the set of time periods while number of vender is more than number of customer and the number of customer is more than the number of time periods. The example of axial three-dimensional problem is the assignment of jobs to workers and machines while number of job is less than number of worker and the number of worker is less than the number of machine. More details in multi-dimension assignment problem can be seen in Pierskalla [37]. Multi-period assignment problem is the part of three-dimensional problem. This problem is assigning agents to changing tasks over the planning periods. An example of this problem is multi-period machine assignment proposed by Zhang and Bard [13]. This paper presented the weekly facilities planning problem at mail processing and distribution center. Those facilities can be used to cancel, sort and sequence the mail. The demand of this problem is the number of machines required for each jobs during the day and planners try to find out the sequence of operations for each machine. A model of Franz and Miller [12] is one of the multi-period assignment problems in which each medical resident is assigned to a variety of operations over many periods. In this model, planning horizon is twelve months and the agents are separated into two types: 2nd resident and 3rd resident. There are many types of operation such as orthopedics, pediatrics and ER. Planner has to decide that each doctor should be assigned to what operation in what months. Other research in multi-period assignment problem can be seen in Mahar, et al. [38], Miller and Franz [34] and Romeijn and Romero Morales[39].

Solution approaches for assignment problems

There are many solution approaches in this area. The mathematical modeling is the popular approach for developing and describing the model. However, many problems in this area is too complex for solving by mathematical model, so many researchers will develop the heuristics for solving the model or try to modify or relax some conditions for reducing the complexity of the problem. Heimerl and Kolisch [40] proposed the model of assigning multi-skill internal and external resources to task while considering skill level and knowledge depreciation. This model is the nonlinear mathematical model which has the objective function to minimize the cost for performing jobs. This model used COIN-OR's Ipopt, which is the open sources program for solving the large-scale nonlinear optimization, for solving the model. The experimental study was done for studying the impact of shape of the learning curve, company skill level target, knowledge depreciation and internal costs. Caron, et al. [16] considered an assignment problem with seniority and job priority constraints with the objective to maximize the number of assigned person. This paper extends the classic assignment problem by considering two types of side constraints: seniority constraint and priority constraint. Both types of constraints appear in many daily scheduling problems in the hospitals. They solve the problem by applying greedy heuristic and use scaling approach. Lieshout and Volgenant [41] developed a branch and bound algorithm for solving the singly constrained assignment problem (SCAP), which is a linear assignment problem with one side constraint. This problem is NP- complete and the new branch and bound algorithm was developed. The new branch and bound algorithm is based on a depth first strategy and lower bounds are obtained by using Lagrangren relaxation. The problem of Foulds and Wilson [35], which designs the tool carousels for flexible manufacturing systems, also applies branch and bound algorithm for solving the model. Punnen and Aneja [42] developed the model for solving resource-constrained assignment problem (RCAP), which is strongly NP-complete. They studied the paper of Mazzola and Neebe [15], which developed the heuristic for RCAP, and found that although this heuristic performed well on the test problems they considered, it was observed to perform poor in some rage of parameter in the RCAP model. Then, they developed a tabu-search-based heuristic algorithm to solve the resource-constrained assignment problem which performs better in cases where the algorithm of Mazzola and Neebe[15] produced poor quality solution. Finally, they suggested using both of these heuristics for producing the good solution. Miller and Franz [34] developed the constrained multi-period assignment problem and Binary-Rounding Heuristic (BRH) is applied for solving this problem. Zhang and Bard [13] considered multi-period machine assignment problem in US mail processing and distribution centers. The mathematical model is developed and construction algorithm and heuristic approach are presented to solve the model.

2.2 Multi-period multi-site assignment problem

The proposed problem is extended from the multi-period multi-site assignment problem. In academic view, this kind of problems are mostly found in the problem of emergency resource allocation [19-21, 31, 32] and the problem of health staff scheduling [17, 18, 22, 23, 28]. An emergency resource allocation is a problem of allocating resources such as health staff, equipments and medical supplies from possible depots or responds units to disaster sites in the disaster situations such as earthquakes, floods or hurricanes while health staff scheduling is a problem of assigning physician, nurses or aides to hospitals, wards or shifts. Both problems consider allocating or assigning resources to suitable shifts or sites as in the proposed problem and many of them consider multiple resource types.

2.2.1 Emergency resource allocation problem

For emergency resource allocation, their resources can be one or more than one type and time period for planning can also be one or more than one period. The performance of allocation processes and decisions in a few days after disaster strikes is an important key to reduce the number of fatalities [32]. Zhang et al. [19] proposed the model and algorithm for allocating multiple resources from emergency depots to disaster sites to fulfill all demands. Ozdamar et al. [20] proposed emergency logistics planning model in natural disaster, which consists of two decisions: allocating multiple commodities to the disaster sites and scheduling the vehicles. Tzeng et al. [21] designd a relief-distribution model for distributing relief items from collection points to transfer candidate depots and relief demand points. More details of this problem can be seen in Caunhye et al. [43] reviewing the optimization models in emergency logistics, Altay and Green Iii [44] surveying the existing OR/MS literatures in disaster operations managements and Fiedrich et al. [32] providing the definitions of core components in Emergency resource allocation problem.

2.2.2 Health staff scheduling problem

For health staff scheduling, a scope of allocation and assignment can be limited in one ward/department [22-27], in many wards/departments in one hospital [28] or in many hospitals [17, 18, 29]. Most researchers have focused on nurse scheduling more than physician scheduling [45] and both staff are mostly considered separately. Ernst et al. [45] reviewed a problem of staff scheduling and rostering in which health staff scheduling is included. Trivedi and Warner [28] proposed algorithm for allocating available float pool of nursing personnel to various inpatient units in a hospital. The nursing personnel are divided into three types which are registered nurses, licensed practical nurses and aides and substituting of those staff is allowed with different performance. Aickelin and Dowsland [22, 24] and Dowsland [25] also considered the problem of nurse scheduling. In their models, there are sets of shift pattern and nurses are divided into many grades in which higher graded nurses can substitute nurses in lower grades. The decision is to assign nurses to shift patterns, which provides the different penalty cost in each shift pattern. For physician scheduling, Carter and Lapierre [23] studied the problem of physician scheduling in emergency room while Goyal and Yadav [29] developed mathematical model and heuristic for allocating physicians to various medical institutions. Other health staff scheduling models can be seen in [17, 18, 26, 27, 46-49].

2.2.3 Solution approaches

There are many solution approaches for the multi-period multi-site assignment problem. Caunhye, Nie et al. [43] reviewed the problem of emergency logistics including solution approaches and techniques used in these models. Tzeng, Cheng et al. [21] developed mathematical model for planning relief delivery and then transformed it to fuzzy multi-objective linear programming for making decision. Fiedrich, Gehbauer et al. [32] applied simulated annealing and tabu search algorithm for allocating resources after earthquake disasters. Zhang, Li et al. [19] proposed 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 technique to assign resource to secondary disaster point. For health staff scheduling problem, Trivedi and Warner [28] proposed modified branch and bound algorithm for allocating float nurses. A problem of Costa Filho, Rivera Rocha et al. [18] concerning human resource allocation was modeled as constraint satisfaction problem and used backtracking search algorithm to find the solution. Others developed algorithms based on mataheuristic concept, for example, ant colony optimization [17], genetic algorithm [22, 24, 27], memetic approach [26] and tabu search algorithm [25, 50].

2.3 Summary

Over the past many years, the classic assignment problem has been extended and many variations of the assignment problem have been proposed. Multi-period multi-site assignment problem is one extension of an assignment problem, which is mostly found in the problem of emergency resource allocation and health resource scheduling. Most of these studies mainly focus on allocating or assigning resources to working sites or shifts. However, they usually consider assigning only one type of resource and if they consider multiple resources types, they do not concern joint requirement of resource types.

For another kind of assignment problem related to the considered problem is the generalized assignment problem (GAP) in which there are many agents who has own capacity for doing tasks and they can do many tasks as long as they have enough capacity. Also, although models in this problem consider multiple resources assignment and their resources can be assigned to many tasks, most models have the condition that each task must to be assigned to only one resource. There is only a model of Alidaee et al. [9] which studied deeply in assigning multiple resources to tasks. However, they have the different objective and do not consider a dimension of multiple periods and multiple sites as in our models.

Although this kind of problem is in the focus of many researchers and have been extended in many areas, there are no any models considering multi-period multi-site assignment problems with joint requirement of multiple resource types as in the considered model.

CHAPTER III
MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM WITH JOINT REQUIREMENT OF TWO RESOURCE TYPES WITHOUT RELOCATION OF RESOURCES (MM-J2-NoRe)

In this chapter the mathematical model and heuristic for multi-period multi-site assignment problem with joint requirement of two resource types without relocation of resource are developed. In this model, the number of resource type is limited to 2 and all resources are not allowed to be rotated.

The remainder of this chapter is organized as follows. In Section 3.1, the problem description is presented. The mathematical model and heuristic algorithm are described in Section 3.2 and 3.3 respectively. Then, the computational experiment is shown in Section 3.4. Finally, in Section 3.5, the conclusion is done.

3.1 Problem description

In this model, the resources are divided into two types (Resource type 1 and Resource type 2) and in each type there are many resources (j = 1, 2,..., J). There are many sites (s = 1, 2,..., S) and periods (p =1, 2,..., P) and in each period, there are many tasks (i =1, 2,..., I). The resource can be assigned to only one site and done only one task per period. Tasks may require one or two resource types for operations as shown in Fig.III-1(B). Task 1 requires two resource types (Resource type 1 and Resource type 2) while Task 2 requires only one resource type (Resource type 2). Only qualified resources can do tasks and task is done only when joint requirements of resources are satisfied.

The decision is to allocate resources to sites and assign resources to tasks to maximize total profit which is calculated from total benefit and total operation cost. Total benefit is calculated from benefit from executed tasks in each period. Total operation cost is calculated from cost of assigning each resource to site.

All details of model can be illustrated in Fig.III-1

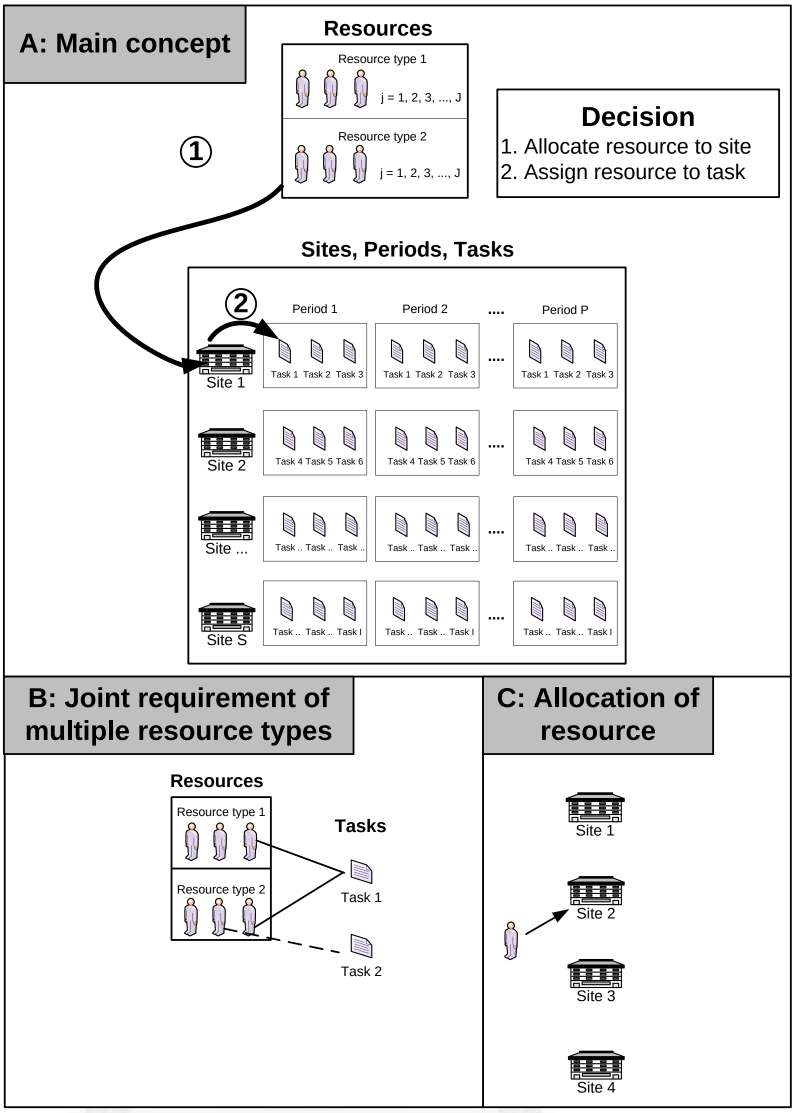


Figure III-1 Characteristics of the MM-J2-NoRe model

When comparing with Fig.I-1 in the Section1.2 (problem description), which is the complete proposed model, the number of resource type is reduced to two and the dimension of date is neglected. The decision is still to allocate resources to sites and assign resources to task; however, each resource is allocated to only one site without relocation as seen from Fig.III-1(C).

We start developing mathematical model and heuristic from this problem because this is the most compact model which has the characteristic of multi-period multi-site assignment problem and demonstrates the effect of joint requirement. It is affiliated for studying the problem when the joint requirement of resource is added and eases for developing mathematical model and preliminary heuristic.

3.2 Mathematical model

From the problem description, a mathematical model can be written as follows.

Index

i = index for tasks; $i \in \{1, 2, 3, \ldots, I\}$

j = index for resources; $j \in \{1, 2, 3, \ldots, J\}$

p = index for periods; $p \in \{1, 2, 3, \ldots, P\}$

s = index for sites; $s \in \{1, 2, 3, \ldots, S\}$

Set

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

Parameters

g^1_{jpi} =1 if the resource j in type 1 is qualified to do task i in period p.

= 200 otherwise. [Big M value]

g^2_{jpi} =1 if the resource j in type 2 is qualified to do task i in period p.

= 200 otherwise. [Big M value]

b^1_{pi} = 1 if task i in period p requires resource type 1.

= 0 otherwise.

b^2_{pi} = 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^1_{js} = operation cost when resource j in type 1 is assigned to site s.

C^2_{js} = operation cost when resource j in type 2 is assigned to site s.

Decision variables

Y^1_{jpi} = 1 if resource j in type 1 is assigned to task i in period p.

= 0 otherwise.

Y^2_{jpi} = 1 if resource j in type 2 is assigned to task i in period p.

= 0 otherwise.

Z^1_{js} = 1 if resource j in type 1 is assigned to site s.

= 0 otherwise.

Z^2_{js} = 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

Maximize total profit = $$\sum_{p=1}^{P}\sum_{i=1}^{I} B_{pi}W_{pi} - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{s=1}^{S} C^1_{js}Z^1_{js} - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{s=1}^{S} C^2_{js}Z^2_{js} \tag{3.1}$$

Constraints

Qualification constraint [resource type 1]:

$$\sum_{i=1}^{I} g^1_{jpi}Y^1_{jpi} \le 1 \;;\forall j \in \{1,\dots,J\},\, p \in \{1,\dots,P\} \tag{3.2A}$$

Qualification constraint [resource type 2]:

$$\sum_{i=1}^{I} g^2_{jpi}Y^2_{jpi} \le 1 \;;\forall j \in \{1,\dots,J\},\, p \in \{1,\dots,P\} \tag{3.2B}$$

Location constraint [resource type 1]:

$$\sum_{s=1}^{S} Z^1_{js} = 1 \;;\forall j \in \{1,\dots,J\} \tag{3.3A}$$

Location constraint [resource type 2]:

$$\sum_{s=1}^{S} Z^2_{js} = 1 \;;\forall j \in \{1,\dots,J\} \tag{3.3B}$$

Joint requirement constraint [resource type 1]:

$$\sum_{j=1}^{J} g_{jpi}^{1} Y_{jpi}^{1} = b_{pi}^{1} W_{pi} \;;\forall i \in \{1,\ldots,I\},\, p \in \{1,\ldots,P\} \tag{3.4A}$$

Joint requirement constraint [resource type 2]:

$$\sum_{j=1}^{J} g_{jpi}^{2} Y_{jpi}^{2} = b_{pi}^{2} W_{pi} \;;\forall i \in \{1,\ldots,I\},\, p \in \{1,\ldots,P\} \tag{3.4B}$$

Available task constraint [resource type 1]:

$$Z_{js}^{1} \geq Y_{jpi}^{1} \;;\forall j \in \{1,\ldots,J\}\, p \in \{1,\ldots,P\}, s \in \{1,\ldots,S\}, i \in I_{ps} \tag{3.5A}$$

Available task constraint [resource type 2]:

$$Z_{js}^{2} \geq Y_{jpi}^{2} \;;\forall j \in \{1,\ldots,J\}\, p \in \{1,\ldots,P\}, s \in \{1,\ldots,S\}, i \in I_{ps} \tag{3.5B}$$

The objective function, Eq. (3.1), maximizes the total profit, which is calculated from benefit and operation cost. Eqs. (3.2A) and (3.2B) enforce that only qualified resources can do tasks and each resource is assigned to only one task per period. Eqs. (3.3A) and (3.3B) enforce that each resource must be assigned to only one site. Eqs. (3.4A) and (3.4B) state that only qualified resources can do tasks and tasks can be done when all requirements of resource are satisfied. Finally, Eqs. (3.5A) and (3.5B) indicate that resources can do only tasks in the site where they are assigned.

Next section, a developed heuristic algorithm is described.

3.3 Heuristic (Heu-1)

In this section, a developed heuristic is presented. The algorithm is separated 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 site and assigning resources to tasks.

Part1: Generating an initial solution.

This part is 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.

Index

i = index for tasks; $i \in \{1,2,3,\ldots,I\}$

j = index for resources; $j \in \{1,2,3,\ldots,J\}$

Parameters

g_{ji}^{1} =1 if the resource j in type 1 is qualified to do task i.

= 200 otherwise. [Big M value]

g_{ji}^{2} =1 if the resource j in type 2 is qualified to do task i.

= 200 otherwise. [Big M value]

b_{i}^{1} = 1 if task i requires resource type 1.

= 0 otherwise.

b_{i}^{2} = 1 if task i requires resource type 2.

= 0 otherwise.

B_{i} = benefit when task i is executed.

Decision variables

Y_{ji}^{1} = 1 if resource j in type 1 is assigned to task i.

= 0 otherwise.

Y_{ji}^{2} = 1 if resource j in type 2 is assigned to task i.

= 0 otherwise.

W_{i} = 1 if task i is executed.

= 0 otherwise.

Objective function

Maximize total profit = $\sum_{i=1}^{I} B_i W_i$ (3.6)

Constraints

Qualification constraint [resource type 1]:

$$\sum_{i=1}^{I} g_{ji}^{1} Y_{ji}^{1} \leq 1 \; ; \forall j \in \{1,\dots,J\} \tag{3.7A}$$

Qualification constraint [resource type 2]:

$$\sum_{i=1}^{I} g_{ji}^{2} Y_{ji}^{2} \leq 1 \;\; ; \forall j \in \{1,\dots,J\} \tag{3.7B}$$

Joint requirement constraint [resource type 1]:

$$\sum_{j=1}^{J} g_{ji}^{1} Y_{ji}^{1} = b_i^{1} W_i \; ; \forall i \in \{1,\dots,I\} \tag{3.8A}$$

Joint requirement constraint [resource type 2]:

$$\sum_{j=1}^{J} g_{ji}^{2} Y_{ji}^{2} = b_i^{2} W_i \; ; \forall i \in \{1,\dots,I\} \tag{3.8B}$$

The decision is reduced to only assign resources to tasks (Y_{ji}^{1}, Y_{ji}^{2} and W_i). Terms of objective function and constraint are also reduced as shown in the Eqs. (3.6)-(3.8). Each sub-problem is calculated by CPLEX to find the initial solution.

Part2: Improving solution

This process is to move resources to new sites and assign resources to tasks. 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 will be 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.III-2.

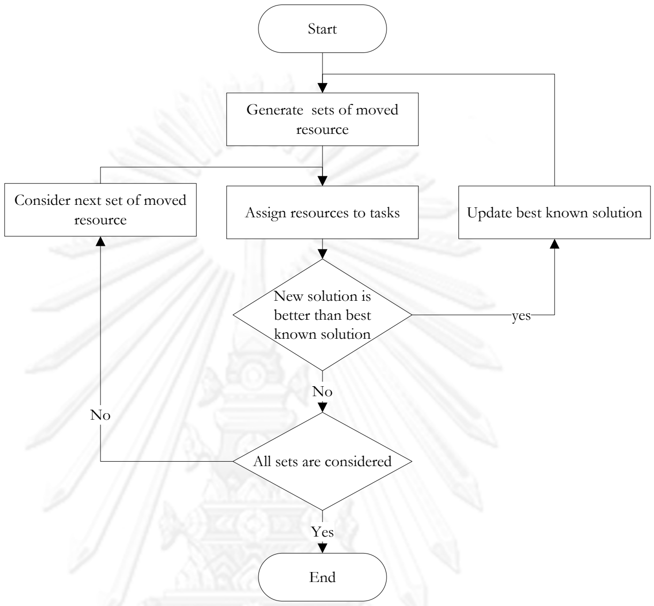


Figure III-2 Process of improving solution

The algorithm starts from generating sets of moved resource, which consist of resources having a potential to improve objective function when they are assigned to another sites. In 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, the only one set of resource 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 resources

A set of moved resources consists of two elements: destination site and selected resources. The destination site is the site to which 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: SetId(m) = {Destination Site Id(s)| Selected resource Id(j) from Type 1, Selected resource Id(j') 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.
    - Group 1: Resource that is idle (resource that is not assigned to any task) and is in the destination site.
    - Group 2: Resource that is idle and is not in the destination site.
    - Group 3: Resource that is assigned to some tasks and is in the destination site.
    - Group 4: 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.

- Group 1: a selected resource is selected randomly.
- Group 2: 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).
- Group 3 and Group 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.III-3.

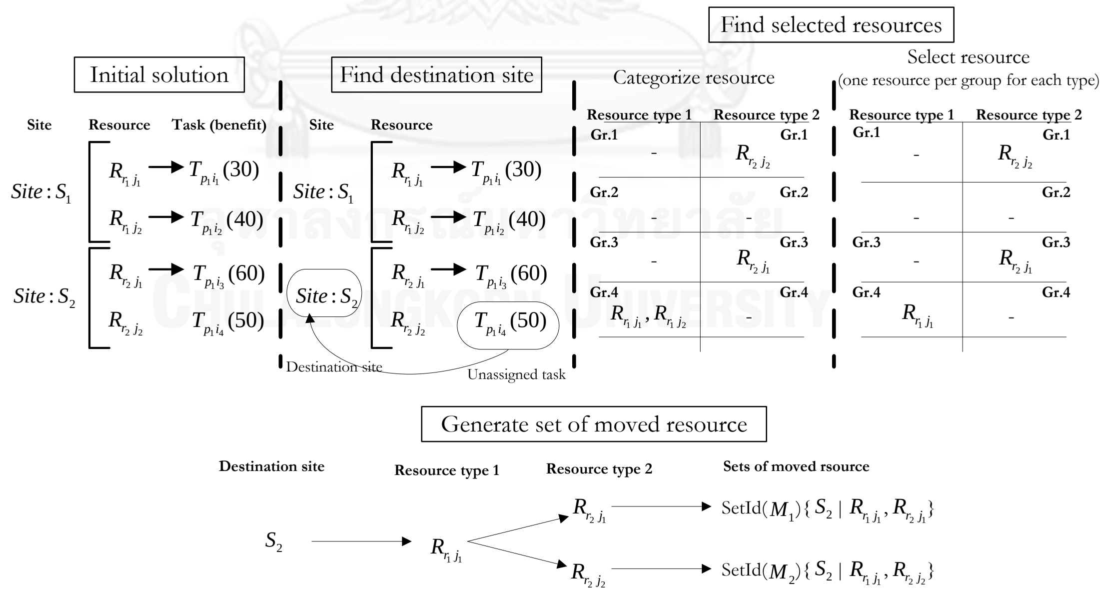


Figure III-3 Example of algorithm for generating sets of moved resource

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 type 1 and 2 and all resources can do this task. For resource type 1(r_1), resource $R_{r_1 j_1}$ 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 cost $C^{r_1}_{j_1 s_1}$ and $C^{r_1}_{j_1 s_2}$ of resource $R_{r_1 j_1}$ are 15 and 20 while $C^{r_1}_{j_2 s_1}$ and $C^{r_1}_{j_2 s_2}$ 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 group 1 and 3 so both resource $R_{r_2 j_1}$ 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 resources can be generated as follows: SetId(M_1){ $S_2 | R_{r_1 j_1}, R_{r_2 j_1}$} and SetId($M_2$){ $S_2 | R_{r_1 j_1}, R_{r_2 j_2}$}.

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 resource. 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.III-4.

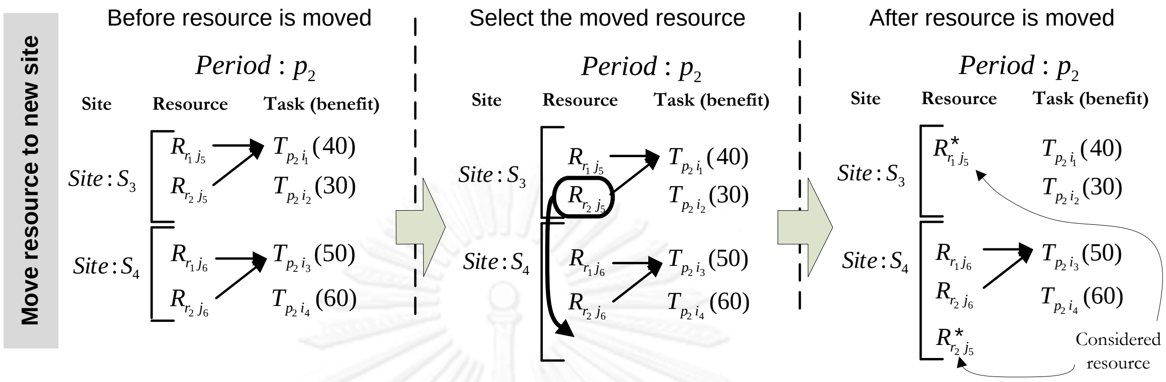


Figure III-4 Example of situation after the resource is moved

In Fig.III-4, 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.III-5, 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.III-5, the considered resource is $R_{r_2 j_5}$ (moved resource). From the concept of greedy search algorithm, task $T_{p_2 i_4}$ which is unassigned task having highest benefit is considered. Assume task $T_{p_2 i_4}$ requires two resource types and resource $R_{r_1 j_6}$ and $R_{r_2 j_5}$ can do this task. Resource $R_{r_1 j_6}$ and $R_{r_2 j_5}$ are assigned to task $T_{p_2 i_4}$ and resource $R_{r_2 j_6}$ is available because of cancelling task $T_{p_2 i_3}$ 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.

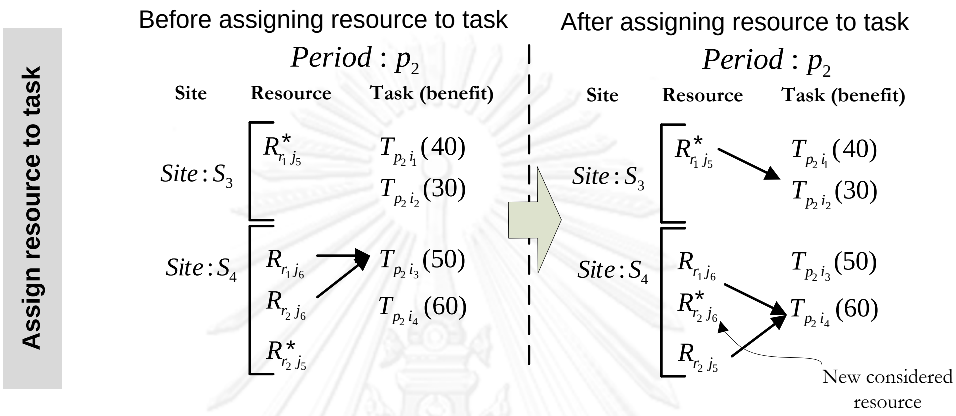


Figure III-5 Example of situation after the resource is assigned to task

In conclusion, the algorithm starts from generating an initial solution by allocating all resources to sites which have lowest operation cost and assigning them to tasks in those sites. From this mechanism, total operation cost should be low. However, many tasks may not be done and total benefit may be low because of mismatching between requirement of tasks and qualification of resources. The next step is to improve the solution by moving resources to other sites to do more tasks to make more profits. The algorithm for generating sets of moved resources will find resources that have the opportunity to do more tasks and make more profit when they are allocated to other sites and then the algorithm for assigning resources to tasks tries to assign resources to unassigned tasks to get as much benefit as possible. The algorithm will be run iteratively until all sets of moved resources are considered and cannot improve objective function.

3.4 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 III-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.

Table III-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.

The results of the experiment are illustrated in Fig.III-6 and Fig.III-7. Figure III-6 shows the computational time and optimal gap of ProbA1 to ProbA4 while Figure III-7 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 [(solution of CPLEX) – (solution of heuristic)] *100 / (solution of CPLEX).

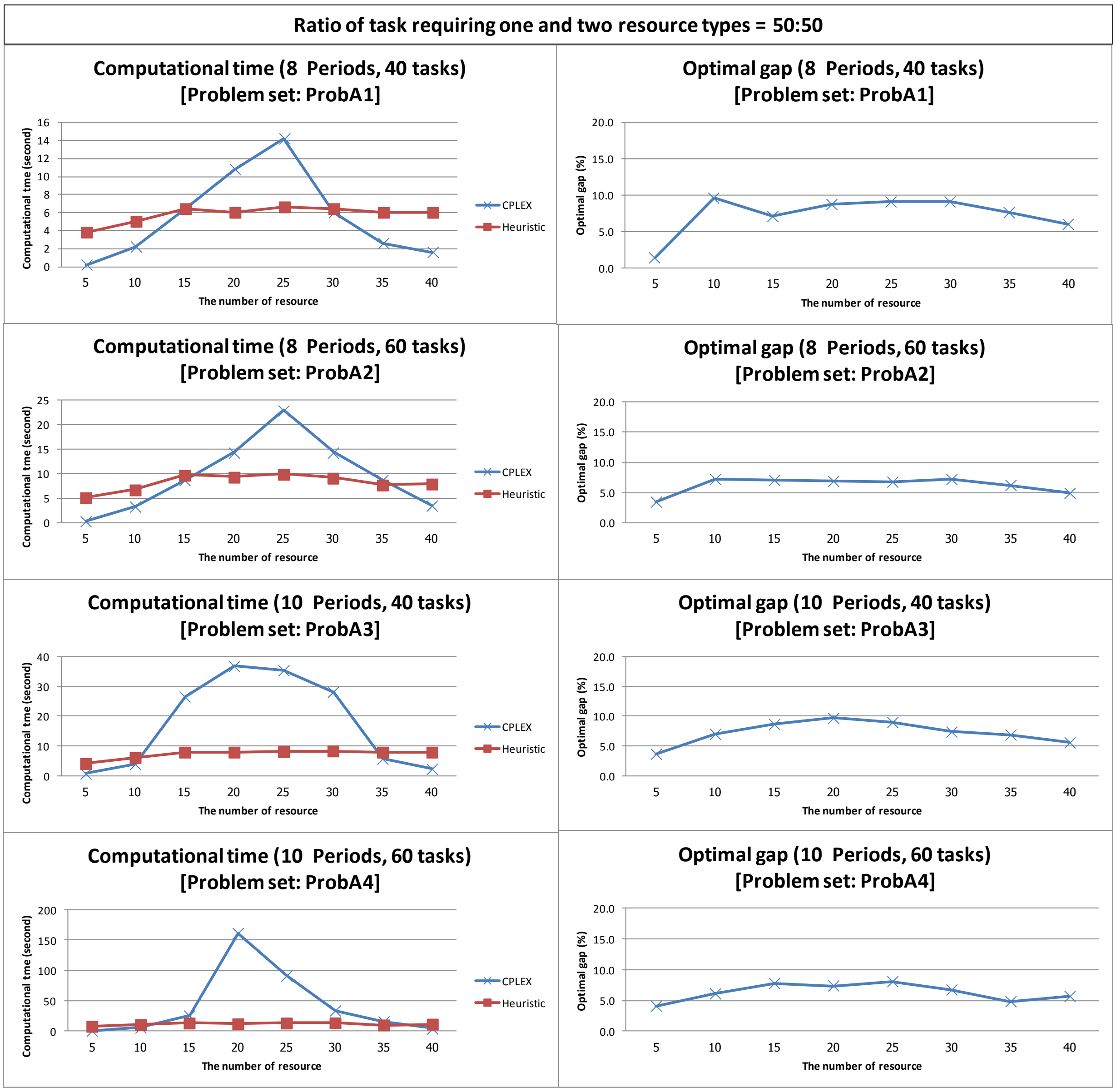


Figure III-6 Computational time and optimal gap of the first experiment [Ratio 50:50]

The finding from the results in Fig.III-6 and Fig.III-7 can be described in two aspects: the complexity of the problem and the efficiency of the developed algorithm.

The result shows that, from Fig.III-6 and Fig.III-7, 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.III-6, 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 [51] 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.III-6 (ProbA4), the number of resource that makes the highest complexity is 20.

When focusing on the efficiency of the algorithm, the developed heuristic method 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 III-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 III-2 shows that, at the highest point where problem is the most complex, the developed heuristic can find good solutions in a short time comparing with CPLEX.

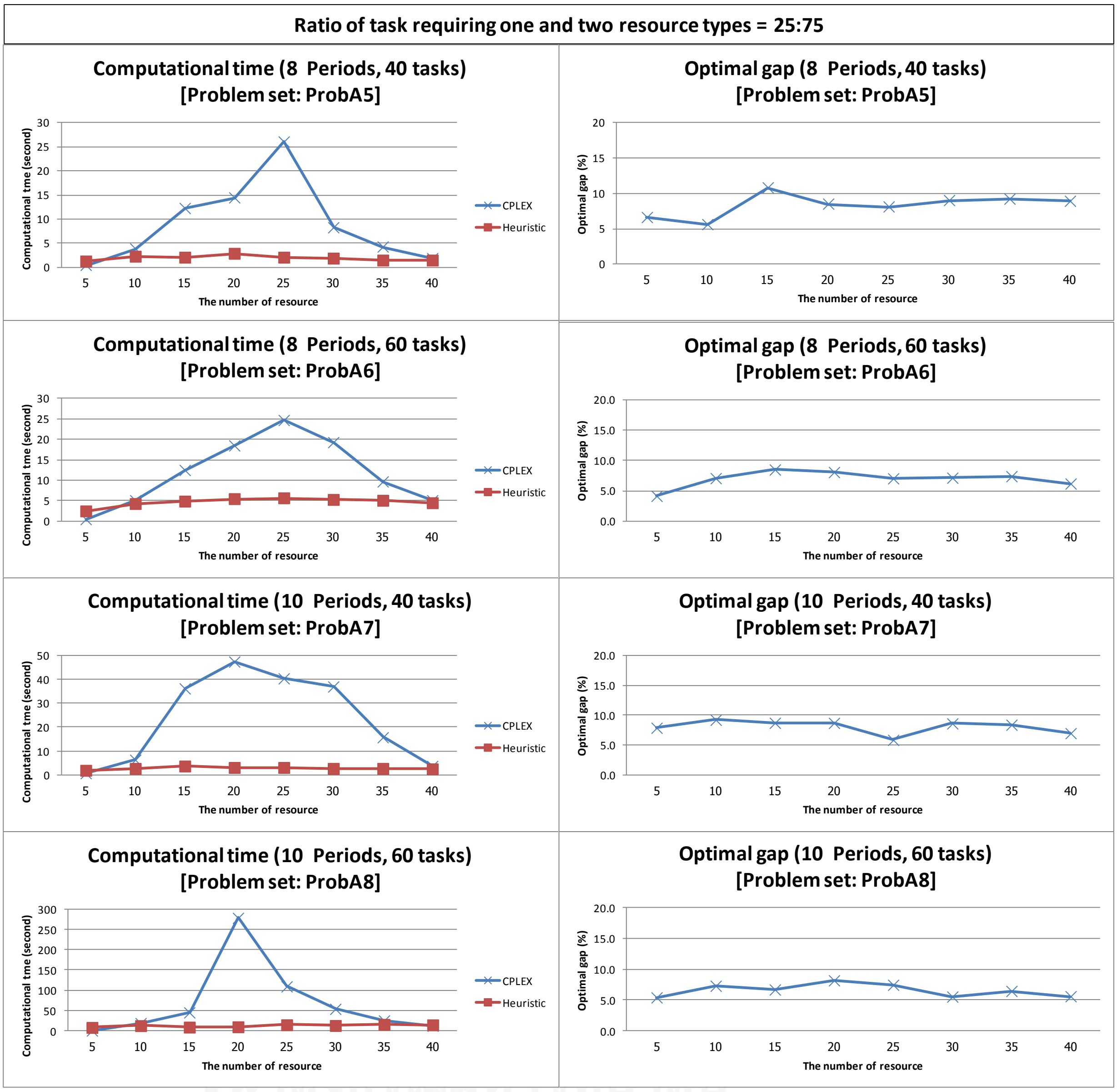


Figure III-7 Computational time and optimal gap of the first experiment [Ratio 25:75]

Table III-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 have to 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 joint of two resource types. On the other hand, if 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.III-6 and ProbA8 of Fig.III-7, the computational time of CPLEX of problems in Fig.III-7 whose ratio is set to 0.25:0.75 is higher than the computational time of problems in Fig.III-6 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 developed. 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 III-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 is fixed to 10, 5 and 60. The number of resource is set to 15 for the first group and 20 for the second group.

Table III-3 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 III-4 and the computational time and optimal gap are illustrated in Fig.III-8. 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 III-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

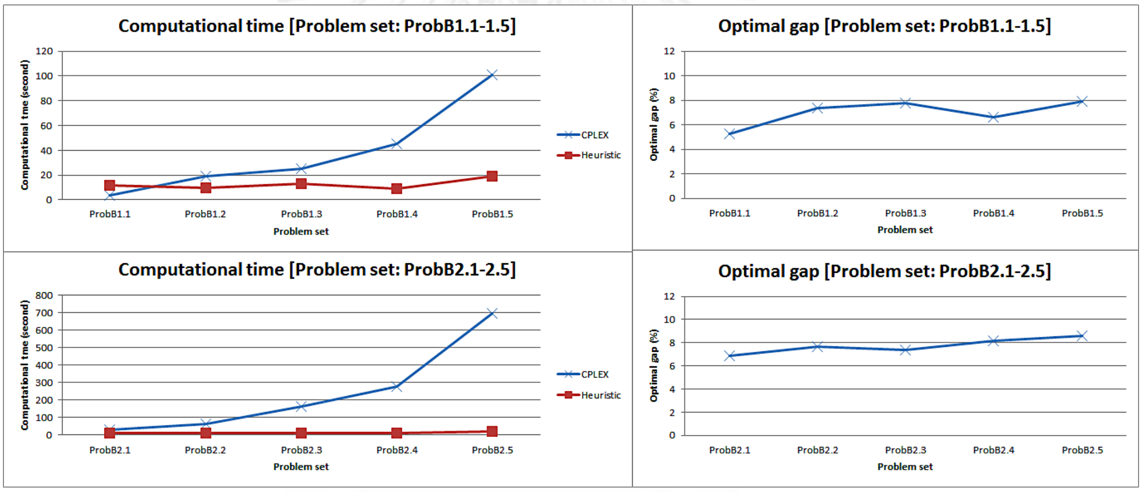


Figure III-8 Computational time and optimal gap of the second experiment

In conclusion, the experiment shows that the computational time of CPLEX dramatically increases when ratio of task requiring two resource types increase and while other parameters are fixed except the number of resource, there is only one range of the number of resource that make 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.

3.5 Conclusion

The purpose of this chapter is to develop model and heuristic for the multi-period multi-site assignment problem concerning joint requirement of two resource types. The developed heuristic is separated into two parts: finding an initial solution and improving 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 to 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%).

CHAPTER IV
MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM WITH JOINT REQUIREMENT OF MULTIPLE RESOURCE TYPES WITHOUT RELOCATION OF RESOURCES (MM-JM-NoRe)

In this chapter the mathematical model and heuristic for multi-period multi-site assignment problem with joint requirement of multiple resource types without relocation of resource are developed. This model is extended from the model in Chapter III. The number of resource type in this model is not limited to two resource types as in the previous model. The mathematical model for this problem is developed based on model in the previous chapter (Chapter III) and heuristic algorithm is developed based on the concept of Tabu search algorithm and algorithm in the previous section.

The remainder of this chapter is organized as follows. In Section4.1, the problem description is presented. The mathematical model and heuristic algorithm are described in Section 4.2 and 4.3 respectively. Then, the computational experiment is shown in Section 4.4. Finally, in Section 4.5, the conclusion is done.

4.1 Problem description

In this model, the number of resource type is not limited to 2 (r= 1, 2,..., R) and in each type there are many resources (j = 1, 2,..., J). There are many sites (s = 1, 2,..., S) and periods (p =1, 2,..., P) and in each period, there are many tasks (i =1, 2,..., I). The resource can be assigned to only one site and done only one task per period. Tasks may require one or more than one resource types for operations as shown in Fig.IV-1(A). Task 1 requires two resource types (Resource type 1 and Resource type 2) while Task 2 requires three resource types (Resource type 1, Resource type 2 and Resource type 3). For Task 3, only Resource type 4 is required. Only qualified resources can do tasks and task is done only when joint requirements of resources are satisfied.

The decision is to allocate resources to sites and assign resources to tasks to maximize total profit which is calculated from total benefit and total operation cost.

Total benefit is calculated from benefit from executed tasks in each period. Total operation cost is calculated from cost of assigning each resource to site.

All details of model can be illustrated in Fig.IV-1

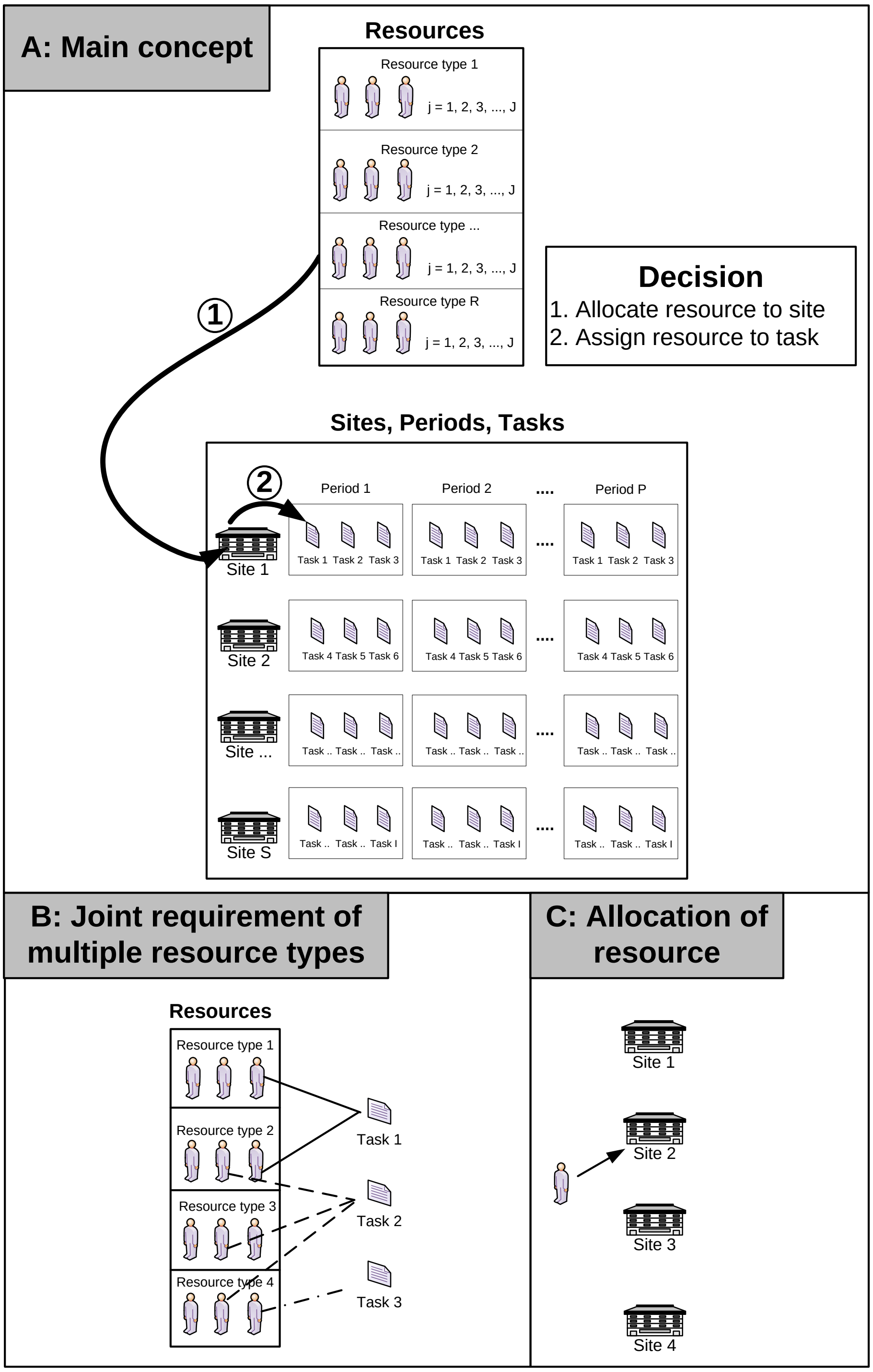


Figure IV-1 Characteristics of the MM-JM-NoRe model

4.2 Mathematical model

From the problem description, a mathematical model can be written as follows.

Index

i = index for tasks; $i \in \{1, 2, 3, \ldots, I\}$

j = index for resources; $j \in \{1, 2, 3, \ldots, J\}$

p = index for periods; $p \in \{1, 2, 3, \ldots, P\}$

r = index for resource types; $r \in \{1, 2, 3, \ldots, R\}$

s = index for sites; $s \in \{1, 2, 3, \ldots, S\}$

Set

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

Parameters

g^{r}_{jpi} =1 if resource j in type r is qualified to do task i in period p.

= 200 otherwise. [Big M value]

b^{r}_{pi} = 1 if task i in period p requires resource type r.

= 0 otherwise.

B_{pi} = benefit when task i in period p is executed.

C^{r}_{js} = operation cost when resource j in type r is assigned to site s.

Decision variables

Y^{r}_{jpi} = 1 if resource j in type r is assigned to task i in period p.

= 0 otherwise.

Z^{r}_{js} = 1 if resource j in type r is assigned to site s.

= 0 otherwise.

W_{pi} = 1 if task i in period p is executed.

= 0 otherwise.

Objective function

Maximize total profit = $\sum_{p=1}^{P}\sum_{i=1}^{I} B_{pi}W_{pi} - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{s=1}^{S} C_{js}^{r} Z_{js}^{r}$ (4.1)

Constraints

Qualification constraint:

$$\sum_{i=1}^{I} g_{jpi}^{r} Y_{jpi}^{r} \leq 1 \; ; \forall r \in \{1,...,R\}, \, j \in \{1,...,J\}, \, p \in \{1,...,P\} \quad (4.2)$$

Location constraint:

$$\sum_{s=1}^{S} Z_{js}^{r} = 1 \; ; \forall r \in \{1,...,R\}, \, j \in \{1,...,J\} \quad (4.3)$$

Joint requirement constraint:

$$\sum_{j=1}^{J} g_{jpi}^{r} Y_{jpi}^{r} = b_{pi}^{r} W_{pi} \; ; \forall r \in \{1,...,R\}, \, p \in \{1,...,P\}, i \in \{1,...,I\} \quad (4.4)$$

Available task constraint:

$$Z_{js}^{r} \geq Y_{jpi}^{r} \; ; \forall r \in \{1,...,R\}, \, j \in \{1,...,J\}, \, p \in \{1,...,P\}, s \in \{1,...,S\}, i \in I_{ps} \quad (4.5)$$

The objective function, Eq. (4.1), maximizes the total profit, which is calculated from total benefits and total operation cost of all resources. Eq. (4.2) enforces that only qualified resources can do tasks and each resource is assigned to only one task per period. Eq. (4.3) enforces that each resource must be assigned to only one site. Eq. (4.4) states that only qualified resources can do tasks and tasks can be done when joint requirements of resources are satisfied, for example, if a task requires resource type 1 and 2, this task can be done (W_{pi} =1) when there are two qualified resources, selected from resource type 1 and 2, assigned to do this task. Each site has different tasks and resources can do only tasks in the site where they are assigned. Eq. (4.5) is used for enforcing this restriction.

4.3 Tabu search heuristic (TS-1)

Tabu search (TS) is a well-known meta-heuristic for solving a combinatorial optimization initiated by Glover [52-54]. A basic process for finding solutions by Tabu search algorithm is roughly divided into three steps: set an initial solution, find neighborhoods and select neighborhood to be new solution. The second and third steps are done iteratively until stopping criteria is met. Efficiency of Tabu search algorithm mainly depends on the structure of neighborhood, Tabu list and long term memory. Good neighborhood lets the algorithm find the best solution in a short time. Because moving to worse solutions is allowed, Tabu list is used to prevent algorithm from cycling or being stuck in a local optimum. Long term memory is usually used to identify the good or bad elements of the solutions or the unvisited regions and then provide the good direction of the next move.

Tabu search algorithm is widely applied in allocation, scheduling and assignment problem [5, 26, 42, 46, 55-57]. For problems whose decision can be divided into many steps as our model, one approach for developing algorithm is to separate the decision into many steps depending on characteristics of the problem and algorithms for each step are developed [27, 45, 46, 58, 59].

For our problem, as described in previous section, the decision of the model is to find the suitable sites (allocation) and suitable tasks (assignment) for resources. To develop algorithm for the considered problem, we separate the decision into two steps: allocating resource to sites and then assigning resources to tasks. We propose a two-step Tabu search algorithm for solving the considered problem: Main Tabu searh algorithm (MTS) for resource allocation in the first step and Sub Tabu search algorithm (STS) for resource assignment in the second step.

A structure of two-step Tabu search algorithm is illustrated in Fig.IV-2. The algorithm starts from generating an initial solution in Main Tabu searh algorithm (MTS). Then, a process of finding all neighborhoods is done. Each neighborhood is a set of resources which should be moved to some sites to provide a better solution. Because getting true objective function of all neighborhoods by using Sub Tabu search algorithm (STS) takes a lot of computational time, we have a process of reducing the number of neighborhood by selecting only some neighborhoods with some criteria to be candidates. After candidate list is generated, Sub Tabu Search Algorithm (STS) will be done to find a solution of resource assignment of each candidate and the true objective function will be calculated. In our STS, we design a specific Tabu list, neighborhood and diversification technique for getting better solution. After all candidates are calculated by STS, the best candidate is selected to be a new initial solution for MTS and the process of updating Tabu list and best known solution in MTS are done. The process is done iteratively until reaching the stopping criteria. The detail of MTS and STS is described as follows.

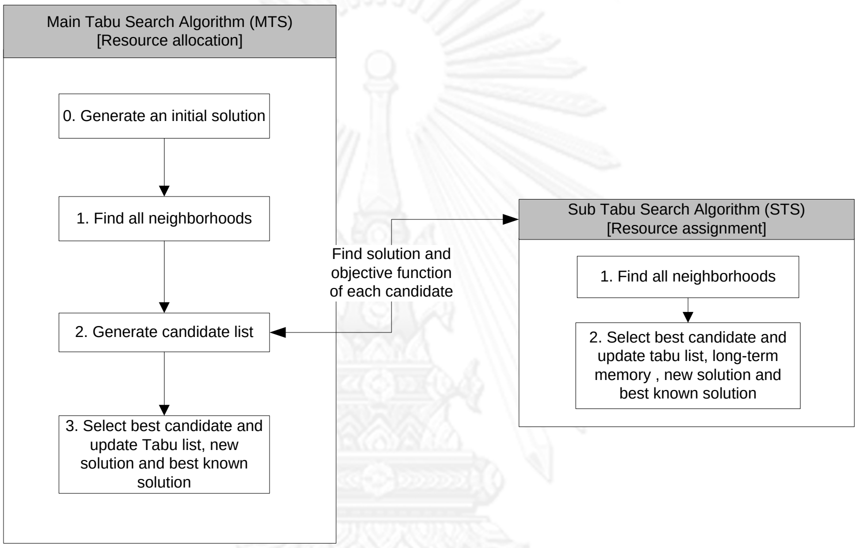


Figure IV-2 Structure of two-step Tabu search algorithm

4.3.1 Main Tabu Search Algorithm (MTS)

Details of Main Tabu Search Algorithm are described as follows.

Generate an initial solution: All resources are assigned to the site that has the lowest operation cost. Then, the problem is split up into many sub-problems. One sub-problem is a problem of one site and one period. Then, CPLEX is used to find an optimal solution of each sub-problem.

Find all neighborhoods: The objective of this step (MTS) is to move resource to the better site. Neighborhoods are generated from moving set of resources from sites to another or, in other words, changing the value of Z_{js}^{r} from 1 to 0 and the value of $Z_{js'}^{r}$ from 0 to 1. The destination site to which these resources are moved is the site where there are unassigned tasks (tasks that nobody does). The moved resources are the resource in each type that can do those unassigned tasks, which selects only one resource per type. For example, in Fig.IV-3, task i_1 in site s_2 is an unassigned task requiring resource type r_2 and r_4. The resource j_1 and j_2 of type r_2 and j_3 and j_4 of type r_4 can do this task. Suppose j_1 and j_3 are selected to move and, in initial solution, j_1 is in s_3 and j_3 is in s_5. Generating neighborhood is to move resource j_1 from site s_3 and j_3 from site s_5 to site s_2 or, in other words, to change the value of $Z^{r_2}_{j_1 s_3}$ and $Z^{r_4}_{j_3 s_5}$ from 1 to 0 and the value of $Z^{r_2}_{j_1 s_2}$ and $Z^{r_4}_{j_3 s_2}$ from 0 to 1. Other neighborhoods can be generated with the same concept which is moving j_1 and j_4 to site s_2, moving j_2 and j_3 to site s_2 and moving j_2 and j_4 to site s_2. This process will be done with all unassigned tasks to generate all neighborhoods.

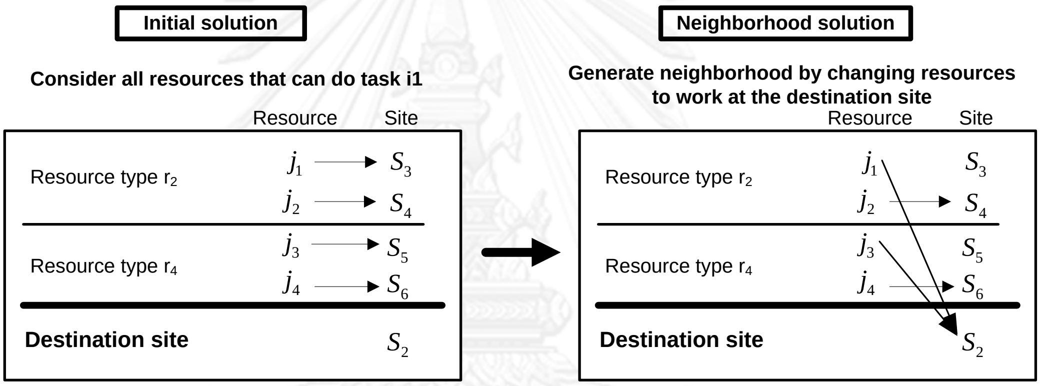


Figure IV-3 Example of generating neighborhood in MTS

Generate candidate list: Because there are a lot of neighborhoods in this step, we reduce the number of neighborhoods by selecting only some neighborhoods. We calculate surrogate objective function of each neighborhood, which consumes short computational time, and select only the best M neighborhoods to be the candidate in the candidate list. If there is more than one candidate generated from one task, only the candidate which has the highest surrogate objective function is considered to be in candidate list.

- Surrogate objective function = Tbg - Tbl + Toc
    - Tbg= Total possible benefit gain, which is the sum of highest benefits of unassigned tasks that each moved resource can do in each period in new site.
    - Tbl = Total benefit lost from moving all resource to new site.

- Toc = Total additional operation cost from moving all resource to new site.

This surrogate objective function is calculated from possible benefits that we will get and lose and additional operation costs that we will pay from changing working sites of resources so this surrogate objective function can accurately approximate the true objective function and is suitable for evaluating the quality of neighborhoods. Then, only good neighborhoods which have the opportunity to improve solutions will be selected to be candidates.

Tabu list: Tabu list is a short term memory used for preventing cycling. In our model, all moved resources from the best neighborhood are added to Tabu list. The resources in Tabu list are not allowed to be moved to other sites for N iterations. This mechanism will prevent an algorithm from sending resources back to the same sites.

Stopping rule: MTS will be run iteratively until reaching the maximum iteration W or the maximum computational time V.

4.3.2 Sub Tabu Search Algorithm (STS)

For each candidate, one or more resources are moved to the new site and then resources in some sites and periods are changed. The algorithm in this step is to assign resources in these sites and periods to unassigned tasks to get as much benefit as possible. Those sites and periods will be calculated by this algorithm one by one until all of them are considered.

An initial solution in this step is the solution from MTS. Details of Sub Tabu Search Algorithm are described as follows.

Find all neighborhoods: Because the objective of this step is to assign resource to unassigned tasks, the neighborhood is generated from selecting some resources to do unassigned tasks ($W_{pi} = 0$). Both available resources (resources which are not assigned to any tasks) and unavailable resources (resources which are assigned to some tasks) are able to be reassigned to do unassigned tasks. A set of resources which provides the minimum benefit lost from reassigning them to do the unassigned task is selected to be the neighborhood. The benefit loss of each neighborhood is calculated from the sum of benefit of all tasks which are cancelled because of changing resources from doing the tasks that they were assigned to doing new unassigned tasks. For example, in Fig.IV-4, an unassigned task i_5 in period p_1 requires resource type r_2, which resource j_3 and j_4 can do, and type r_3, which resource j_5 and j_6 can do. Suppose, in initial solution, j_3 and j_4 do task i_1 and i_2 whose benefit is 100 and 50 respectively while j_5 and j_6 do task i_3 and i_4 whose benefit is 200 and 150 respectively. To generate neighborhood, we select resource j_4 and j_6 to do this unassigned task because they provide the minimum benefit lost from cancelling tasks ($B_{p_1 i_2} + B_{p_1 i_4} = 50 + 150 = 200$). The neighborhood is to change the value of $Y^{r_2}_{j_4 p_1 i_2}$, $W_{p_1 i_2}$, $Y^{r_3}_{j_6 p_1 i_4}$ and $W_{p_1 i_4}$ from 1 to 0 and change the value of $Y^{r_2}_{j_4 p_1 i_5}$, $Y^{r_3}_{j_6 p_1 i_5}$ and $W_{p_1 i_5}$ from 0 to 1. Qualified resources in this step must not only be able to do unassigned tasks but also be in the site which has those unassigned tasks. This step is done with all unassigned tasks to generate all neighborhoods. After having all neighborhoods, the neighborhood which has the highest objective function is selected to be a new initial solution. This process is done iteratively until the objective function does not improve for P iterations.

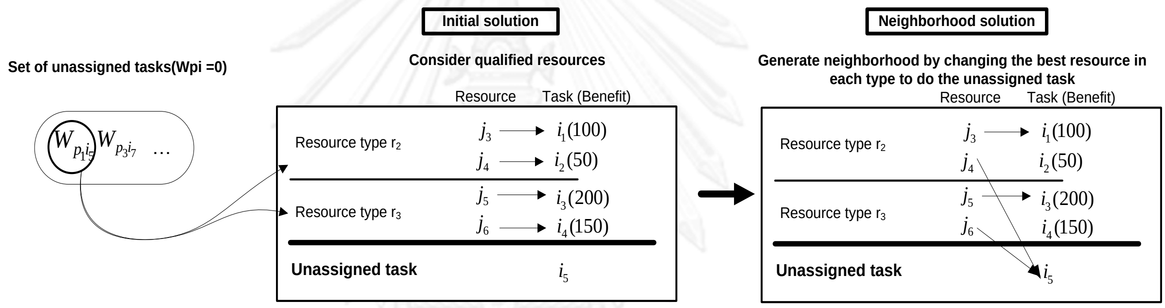


Figure IV-4 Example of generating neighborhood in STS

Tabu list: All changed resources and tasks from the best neighborhood are added to Tabu list. They are not allowed to be changed for Q iterations. This mechanism will prevent an algorithm from assigning the same resources to the same tasks.

Diversification technique: This is a technique in Tabu search algorithm to enable the searching process to move and find neighborhoods in different area of solution space. We collect a frequency of tasks done in each iteration to long term memory. If the objective function of solution does not improve for D iterations, the benefit of each task will be divided by this frequency and the new benefit will be used to find the neighborhood. The process will be done for E iterations and then the long term memory will be reset. From this mechanism, high benefit tasks which are usually done in previous iterations will have less opportunity to be selected to be good candidates and low benefit tasks which are rarely done by any resources will have more opportunity to be selected instead.

In conclusion, the algorithm starts from generating an initial solution by allocating all resources to sites which have lowest operation cost and assigning them to tasks in those sites. From this mechanism, total operation cost should be low. However, many tasks may not be done and total benefit may be low because of mismatching between requirement of tasks and qualification of resources. The next step is to improve the solution by moving resources to other sites to do more tasks to make more profits. The mechanism of neighborhood structure will find the resources that have the opportunity to do more tasks and make more profit when they are allocated to other sites.

Because there are many neighborhoods, selecting only good neighborhoods by using surrogate objective function will greatly reduce the computational time for finding good solutions. After resources are moved to new site, there are only some sites and period whose resources are changed. STS algorithm will be done to assign resources to tasks in these sites and periods. Mechanism in this step tries to assign resources to unassigned tasks to get as much benefit as possible. Tabu list in STS algorithm will prevent the algorithm from assigning the same resources to the same tasks after running for many iterations. When the solutions do not improve for many iterations, diversification technique will be done for diverting the algorithm to focus on unassigned tasks which have lower benefit and are rarely considered.

After improving process is done for many iterations, resources may be back to the same site which is the cause of being struck in a local optimum. The structure of Tabu list in MTS algorithm will prevent this problem by not allowing resources in Tabu list to move back to the same site.

The pseudo-code of the developed Tabu search algorithm (TS-1) can be written as follows.

======================Pseudo-code of TS-1 algorithm=================

<u>Definition of parameters, variables and sets for Main Tabu Search Algorithm</u>

$x_{initial}$ = the initial solution

x_{newsol} = the solution obtained in each iteration

$x_{bestknown}$ = the best known solution

$N_m(x)$ = a set of neighborhood of solution x

$C_m(x)$ = a set of candidate of solution x

$SurroObj(x)$ = a surrogate objective function of candidate x

$Obj(x)$ = an objective function of candidate x

$numCandidate$ = the number of neighborhood in candidate list (M)

$MainTabu\_maxIte$ = the maximum iteration for running MTS (W)

$Tabulist_m$ = a set of resources in Tabu list

Algorithm TS1_1: Main Tabu Search Algorithm (MTS algorithm)

Generate an initial solution: $x_{initial}$;

Set $x_{newsol} = x_{initial}$, $x_{bestknown} = x_{initial}$, $MainTabu\_countIte = 0$;

repeat

Generate neighborhoods: $N_m(x_{newsol})$;

Let $x \in N_m(x_{newsol})$;

Set $countCandidate = 0, C_m(x_{newsol}) = \varnothing$;

while ($countCandidate < numCandidate$)

Find x which has the highest $SurroObj(x)$ and is not in $C_m(x_{newsol})$;

Add x to $C_m(x_{newsol})$;

Set $countCandidate++$;

end while

for each candidate (x) in $C_m(x_{newsol})$ **do**

Use Algorithm TS1_2 to obtain the solution: x';

Delete x in $C_m(x_{newsol})$ and add x' in $C_m(x_{newsol})$;

end for

Initiate a dummy solution x_{dummy} such that $Obj(x_{dummy})$ = int.MinValue;

for each candidate (x') in $C_m(x_{newsol})$ **do**

if $Obj(x') > Obj(x_{dummy})$ **then**

Set $x_{dummy} = x'$;

end if

end for

Set $x_{newsol} = x_{dummy}$;

Update $Tabulist_m$;

if $Obj(x_{newsol}) > Obj(x_{bestknown})$ **then**

Set $x_{bestknown} = x_{newsol}$;

end if

Set $MainTabu\_countIte++$;

until $MainTabu\_countIte = MainTabu\_maxIte$

==

<u>Definition of parameters, variables and sets for Sub Tabu Search Algorithm</u>

y_{ps} = the solution of site s in period p

$y_{ps}^{bestknown}$ = the best known solution of site s in period p

$Benefit(y)$ = the benefit from solution y

$SubTabu\_numUseDivert$ = the number of iteration for using diversification strategy (D)

$SubTabu\_maxUseDivert$ = the duration for using diversification strategy (E)

$SubTabu\_freqOfTask(t)$ = the frequency of task t in long term memory

$Tabulist_s$ = a set of resources and tasks in Tabu list

$SubTabu\_maxIte$ = the maximum iteration for running STS (P)

Algorithm TS1_2: Sub Tabu Search Algorithm (STS algorithm)

Get candidate (x) from $C_m(x_{newsol})$;

Initiate $x' = x$;

for each site and period whose resources are changed **do**

Obtain y_{ps} from x;

Set $y_{ps}^{bestknown} = y_{ps}$;

repeat

Set $SubTabu\_countIte, SubTabu\_countUnimprove = 0$;

Set $countUseDivert = 0$;

Generate neighborhoods: $N_s(y_{ps})$;

for $y' \in N_s(y_{ps})$ **do**

Find $Benefit(y')$;

Set t' = the selected task in neighborhood y';

if $SubTabu\_countUnimprove = SubTabu\_numUseDivert$ or $countUseDivert > 0$ **then**

Set $Benefit(y') = Benefit(y') / FreqOfTask(t')$;

Set $countUseDivert{+}{+}$;

if $countUseDivert = SubTabu\_maxUseDivert$ **then**

for $t \in$ tasks in a considered site and period **do**

Set $FreqOfTask(t) = 1$;

Set $countUseDivert = 0$;

Set $SubTabu\_countUnimprove = 0$;

end for

end if

end if

end for

Initiate a dummy solution y_{dummy} such that $Benefit(y_{dummy})$ = int.MinValue;

for $y' \in N_s(y_{ps})$ do

if $Benefit(y') > Benefit(y_{dummy})$ then

Set $y_{dummy} = y'$;

end if

end for

Set $y_{ps} = y_{dummy}$;

Update $Tabulist_s$ and Long term memory;

if $Benefit(y_{ps}) > Benefit(y_{ps}^{bestknown})$ then

Set $y_{ps}^{bestknown} = y_{ps}$;

Set $SubTabu\_countUnimprove = 0$;

else

Set $SubTabu\_countUnimprove++$;

end if

Set $SubTabu\_countIte++$;

until $SubTabu\_countIte = SubTabu\_maxIte$

Update $y_{ps}^{bestknown}$ to x' ;

end for

return x' ;

==

4.4 Computational experiment

A Tabu search algorithm is tested to evaluate the efficiency of the proposed algorithm to the considered problem. The algorithm was coded in C# 2010 and run on the Windows 7 Ultimate with Intel Core i5-2410M, CPU 2.30GHz and RAM 4GB. We compare our results with solutions from commercial optimization tool (ILOG CPLEX 12.1.0).

Test problems are generated into three different sizes. The first set of problem is a small size problem which takes short computational time. That is, CPLEX can find an optimal solution in a few second. The second set is a medium size problem which takes less than 4,000 seconds to find an optimal solution while the third set, a large size problem, takes more than 4000 seconds.

For the small size problem, the number of resource type is fixed to 2 (ratio of task that requires 1 type and 2 types is set to 25%: 75%). The number of period is set to 3 and 6 while the number of site is set to 5 and 10. A ratio of resource and task is set to 1:2 [6 resources: 12 tasks and 10 resources: 20 tasks] and ratio of resource that can do each task is set to 0.4. Operation cost and benefit are randomized uniformly between 500 to 2,000 and 400 to 4,000 respectively. For each problem set, 10 tests are generated.

For the medium size problem, the experiment is separated into 2 parts. A ratio of resource and task is varied in the first part while the number of resource type and the ratio of task that requires each type are varied in the second part. In the first part, the number of resource is varied from 10 to 16 and the ratio of resource and task is set to 1:2 and 1:4. The number of resource type, period and site are fixed to 2, 12 and 5 respectively. In the second part, the number of resource type is set to 2 and 3. The ratio of tasks that requires each resource type is varied from 0% to 100%. The number of resource, period, site and task are fixed to 14, 9, 5 and 30 respectively and ratio of resource that can do each task is set to 0.4. Operation cost and benefit are randomized uniformly between 2,000 to 10,000 and 400 to 4,000 respectively. For each problem set, 5 tests are generated.

For the large size problem, the number of resource, period, site and task are fixed to 20, 12, 5 and 60 respectively. The number of resource type is set to 4, 6, 8, 10 and 12 and ratio of resource that can do each task is set to 0.4. Operation cost is uniformly randomized between 2,000 to 10,000 for all problems while a benefit is uniformly randomized between 400 to 4,000 for 4, 6 and 8 resource types, 800 to 8,000 for 10 resource types and 1,200 to 12,000 for 12 resource types. The ranges of benefit in each problem size are different to maintain the value of objective function to be positive. For each problem set, 1 test is generated.

The details of all problem sizes are shown in Table IV-1. The first seven columns are the description of tested problems, which is the size of problem, the set of problem, the number of resource, the number of task, the number of period, the number of site and the number of resource type, and the rest are the ratio of tasks that requires each resource type for operations.

Table IV-1 Details of all problem sizes

Problem size	Problem set	Number of resource	Number of task	Number of period	Number of site	Number of resource type	Ratio of each resource type											
							1	2	3	4	5	6	7	8	9	10	11	12
Small problem	S1	6	12	3	10	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	S2	6	12	6	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	S3	10	20	3	10	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	S4	10	20	6	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
Medium problem (Part1)	MA1.1	10	20	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MA1.2	12	24	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MA1.3	14	28	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MA1.4	16	32	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MA2.1	10	40	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MA2.2	12	48	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MA2.3	14	56	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MA2.4	16	64	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
Medium problem (Part2)	MB1.1	14	30	12	5	2	1.00	0	-	-	-	-	-	-	-	-	-	-
	MB1.2	14	30	12	5	2	0.75	0.25	-	-	-	-	-	-	-	-	-	-
	MB1.3	14	30	12	5	2	0.50	0.50	-	-	-	-	-	-	-	-	-	-
	MB1.4	14	30	12	5	2	0.25	0.75	-	-	-	-	-	-	-	-	-	-
	MB1.5	14	30	12	5	2	0	1.00	-	-	-	-	-	-	-	-	-	-
	MB2.1	14	30	9	5	3	1.00	0	0	-	-	-	-	-	-	-	-	-
	MB2.2	14	30	9	5	3	0.60	0.20	0.20	-	-	-	-	-	-	-	-	-
	MB2.3	14	30	9	5	3	0.20	0.60	0.20	-	-	-	-	-	-	-	-	-
	MB2.4	14	30	9	5	3	0.20	0.20	0.60	-	-	-	-	-	-	-	-	-
	MB2.5	14	30	9	5	3	0	0	1.00	-	-	-	-	-	-	-	-	-
Large problem	L1	20	60	12	5	4	0.10	0.20	0.30	0.40	-	-	-	-	-	-	-	-
	L2	20	60	12	5	6	0.05	0.10	0.15	0.15	0.20	0.35	-	-	-	-	-	-
	L3	20	60	12	5	8	0.05	0.05	0.10	0.10	0.15	0.15	0.20	0.20	-	-	-	-
	L4	20	60	12	5	10	0.01	0.04	0.05	0.05	0.06	0.10	0.14	0.15	0.20	0.20	-	-
	L5	20	60	12	5	12	0.01	0.01	0.03	0.03	0.06	0.06	0.09	0.09	0.13	0.13	0.18	0.18

All parameters of Tabu search algorithm are set according to size of the problem. From preliminary experiments, the suitable parameters for STS can be set as follows:

$$P = \text{ROUNDUP}(6*(\text{number of task}),0) \tag{4.6}$$

$$Q = \text{ROUNDUP}\left(\text{MIN}\left(0.7*(\text{number of resource}), 0.1*(\text{number of task})\right), 0\right) \tag{4.7}$$

$$D = \text{ROUNDUP}(0.4*(\text{number of task}),0) \tag{4.8}$$

$$E = 2 \tag{4.9}$$

For MTS, a suitable size of Tabu list (N) to prevent algorithm from being stuck in a local optimum is 3 or 4 depending on the problem size. The number of maximum iteration (V) and candidate (M) affect directly to the quality of the solution. A larger number of V and M increase the opportunity to find better solutions; however, it takes more computational time. V and M are limited to the suitable value according to the problem size and the computational time. All MTS and STS parameters are shown in Table IV-2. The parameters are divided into 2 groups: MTS and STS. In MTS, the Max iteration (W)/time (V), Tabu list (N) and Candidate list (M) is the maximum iteration or maximum time for running MTS, the number of iteration for keeping moved resources in Tabu list and the number of neighborhood in candidate list respectively. In STS, the Max iteration (P), Tabu list (Q) Tricker for Divert (D) and Duration for Divert (E) is the maximum iteration for running STS, the number of iteration for keeping changed resources and tasks in Tabu list, the number of iteration for using diversification technique and the duration for using diversification technique respectively.

Table IV-2 Parameters for all test problems

Problem size	Problem set	Parameters						
		MTS			STS			
		Max iteration (W) /time (V)	Tabu list (N)	Candidate list (M)	Max iteration (P)	Tabu list (Q)	Tricker for Divert (D)	Duration for Divert (E)
Small problem	S1	5 seconds	3	4	8	1	1	2
	S2	5 seconds	3	8	15	1	1	2
	S3	5 seconds	3	8	12	1	1	2
	S4	5 seconds	3	4	24	1	2	2
Medium problem (Part1)	MA1.1	200 iterations	3	3	1	2	2	24
	MA1.2	200 iterations	3	3	1	2	2	29
	MA1.3	200 iterations	3	3	1	3	2	34
	MA1.4	200 iterations	3	3	1	3	2	39
	MA2.1	200 iterations	3	3	1	4	2	48
	MA2.2	200 iterations	3	3	1	4	2	58
	MA2.3	200 iterations	3	3	2	5	2	68
	MA2.4	200 iterations	3	3	2	6	2	77
Medium problem (Part2)	MB1.1	200 iterations	3	3	1	3	1	36
	MB1.2	200 iterations	3	3	1	3	1	36
	MB1.3	200 iterations	3	3	1	3	1	36
	MB1.4	200 iterations	3	3	1	3	1	36
	MB1.5	200 iterations	3	3	1	3	1	36
	MB2.1	80 iterations	3	6	1	3	1	36
	MB2.2	80 iterations	3	6	1	3	1	36
	MB2.3	80 iterations	3	32	1	3	1	36
	MB2.4	80 iterations	3	64	1	3	1	36
	MB2.5	80 iterations	3	170	1	3	1	36
Large problem	L1	100 iterations	4	32	2	5	2	72
	L2	100 iterations	4	32	2	5	2	72
	L3	100 iterations	4	32	2	5	2	72
	L4	100 iterations	4	32	2	5	2	72
	L5	100 iterations	4	32	2	5	2	72

*The value of all parameters in STS is calculated by equation (4.6) – (4.9)

The experiment of small size problems

In the small size problem, 4 problems are generated: S1, S2, S3 and S4. For S1 and S3, there are only few periods but many sites whereas, for S2 and S4, there are only few sites but many periods. The number of resource and task in S3 and S4 is more than in S1 and S2. Time for running MTS for all problems is limited to 5 seconds. The results of the experiment are illustrated in Table IV-3. The second column (#OPT by Tabu) shows the number of optimal solution found by Tabu search algorithm. The average optimal gap is shown in the third column, which is calculated from [(solution of CPLEX) – (solution of Tabu search)] *100 / (solution of CPLEX). The result shows that, for all the test problems (40 tests), 13 optimal solutions are found and the average optimal gap ranges from 0.6 to 4.0.

Table IV-3 The results from experiments of small size problems

Problem set	#OPT by Tabu (10 tests)	Average gap (%)
S1	5	0.6
S2	7	1.5
S3	1	2.7
S4	0	4.0

The experiment of medium size problems [part1]

In the medium size problem [part1], the experiment is separated into two groups and four problems per group are generated: MA1.1 to MA1.4 for the first group and MA2.1 to MA2.4 for the second group. All parameters in both groups are the same except the number of task which is set depending on the ratio of resource and task (1.2 for the first group and 1.4 for the second group). The result of the experiment is shown in Table IV-4. The second column shows the computational time of CPLEX while the third column shows time to find the best solution of Tabu search algorithm. The forth to sixth column show the minimum optimal gap, maximum optimal gap and average optimal gap. The average optimal gap and computational time are plotted in Fig.IV-5. The result shows that the computational time of CPLEX considerably increases when the number and ratio of resource and task increase. Tabu search algorithm can find good solutions in a short time comparing with CPLEX and the quality of the solution remains good when the number and ratio of the resource and task increase.

Table IV-4 The results from experiments of medium size problems [part1]

Problem set	CPLEX time (sec)	Tabu time (sec)	Minimum gap (%)	Maximum gap (%)	Average gap (%)
MA1.1	52	5	2.8	7.9	5.3
MA1.2	48	6	3.3	8.8	6.0
MA1.3	108	33	3.3	7.4	5.1
MA1.4	775	17	2.8	7.5	4.8
MA2.1	66	39	3.3	7.4	5.5
MA2.2	212	49	3.8	6.0	4.5
MA2.3	335	64	4.8	6.7	5.3
MA2.4	953	99	3.7	5.4	4.4

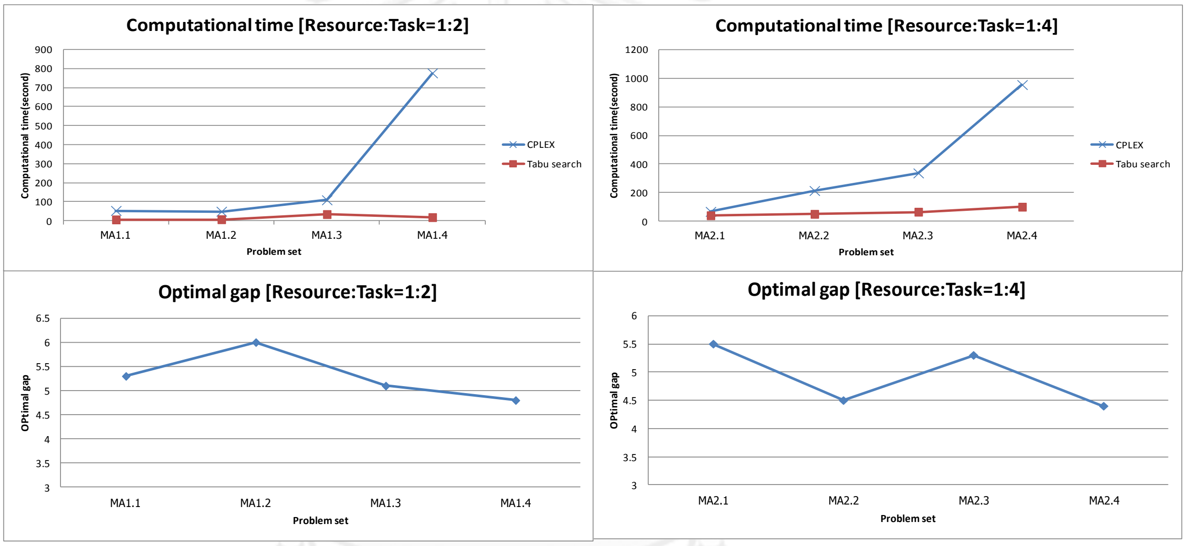


Figure IV-5 Computational time and optimal gap of medium size problem [part1]

The experiment of medium size problems [part2]

In the medium size problem [part2], the experiment is separated into two groups and five problems per group are generated: MB1.1. to MB1.5 for the first group and MB2.1 to MB2.5 for the second group. All parameters in both groups are the same except the number of resource type varied from 2 to 3 and the ratio of task that requires each type varied from 0% to 100%. The result of the experiment is shown in Table IV-5 and the computational time and optimal gap are illustrated in Fig.IV-6. The result shows that the computational time of CPLEX extremely increases when the number of resource type and the ratio increase (for this experiment, the number of resource type just increases by 1). The optimal gap slightly increases when the ratio increases. However, the algorithm can still provide good solutions in a short time comparing with the CPLEX.

Table IV-5 The results from experiments of medium size problems [part2]

Problem set	CPLEX time (sec)	Tabu time (sec)	Minimum gap (%)	Maximum gap (%)	Average gap (%)
MB1.1	2	8	2.7	4.2	3.7
MB1.2	38	13	0.9	6.8	3.7
MB1.3	88	29	2.8	7.5	4
MB1.4	178	38	4.7	7.9	6.1
MB1.5	619	28	5.8	8.7	7.1
MB2.1	1	7	1.53	6.09	4.6
MB2.2	13	11	2.61	10.02	5.6
MB2.3	183	65	4.35	8.28	6
MB2.4	1334	120	5.11	11.35	8.1
MB2.5	2435	370	10.85	18.82	13.9

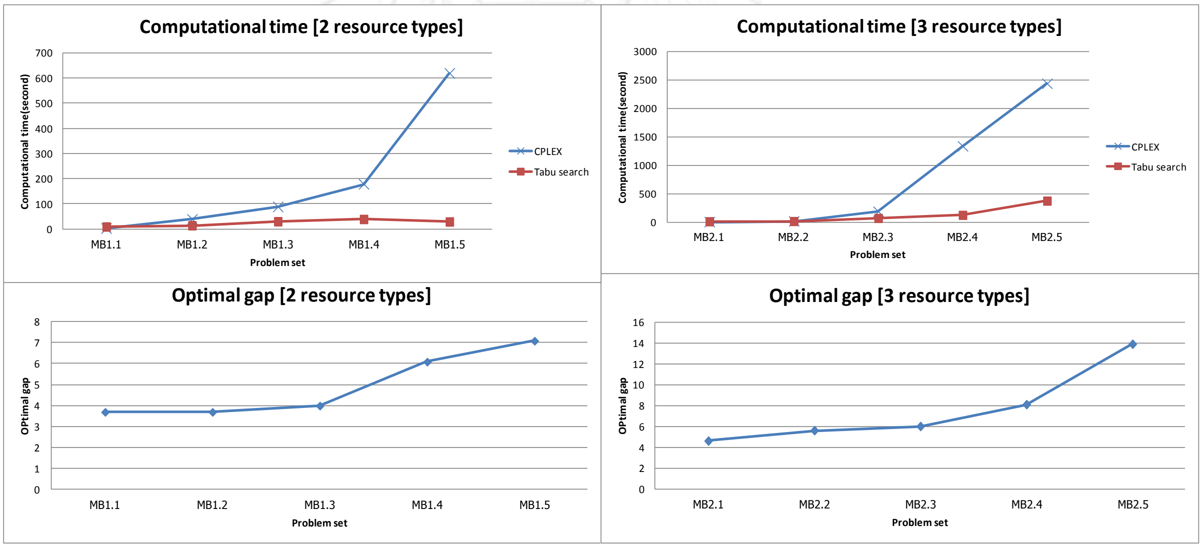


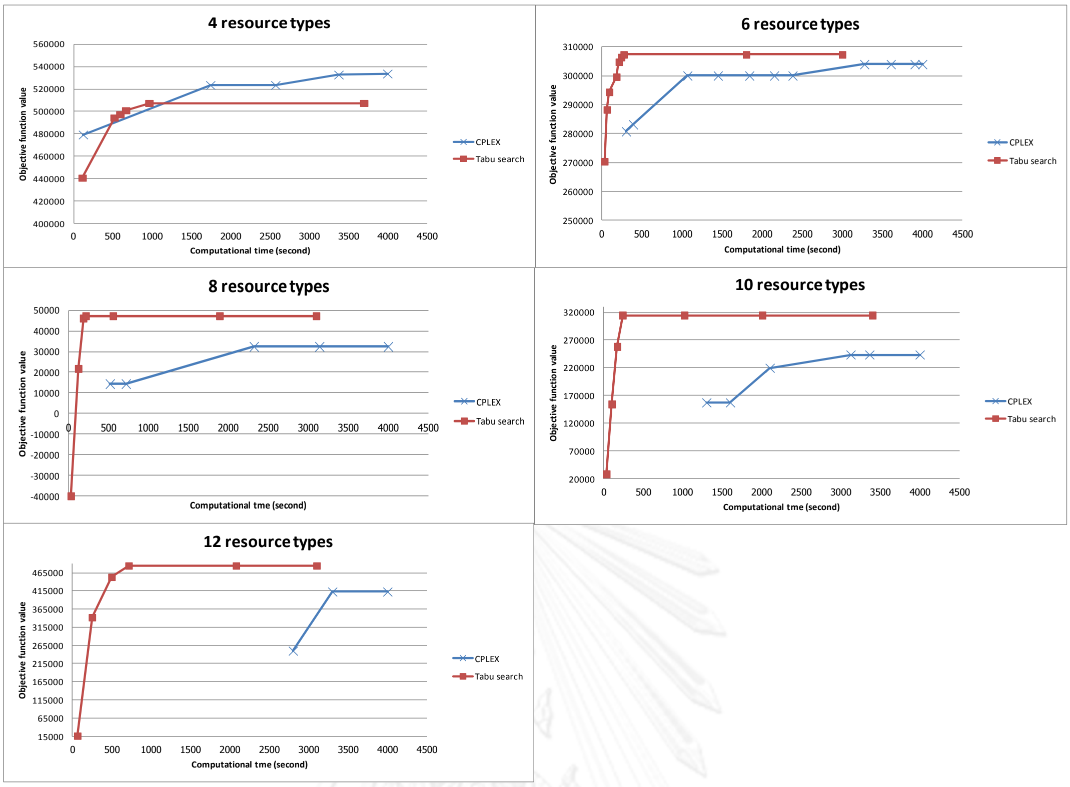
Figure IV-6 Computational time and optimal gap of medium size problem [part2]

The experiment of large size problems

In the large size problem, five problems are generated: L1, L2, L3, L4 and L5. Because finding an optimal solution for this problem set takes a lot of time, the computational time for running CPLEX is limited to 4000 seconds while time for running MTS is limited to 100 iterations and the number of candidate list is fixed to 32. We know from the result of experiment in medium size problem that the computational time or complexity of the problem extremely increases when the number of resource type increases. In this experiment, the number of resource type is varied from 4 to 12, which is a large number comparing with the previous experiment (2 and 3 resource types), to evaluate the efficiency of the algorithm when being applied to high complexity problems. The computational time and optimal gap are shown in Table IV-6. The value of optimal gap in this problem set can be less than zero because running time of CPLEX is limited and solutions from CPLEX can be worse than the solutions from Tabu search algorithm. As shown in the Table IV-6, the optimal gap of four out of five problems is less than zero, which means that four out of five solutions from Tabu search algorithm are better than solutions from CPLEX. Figure IV-7 illustrates the computational time and the best solution found in each time by CPLEX and Tabu search algorithm. Parts of CPLEX line that have no data means that CPLEX cannot find any feasible solution. As can be seen, when the number of resource type increases, time to find a feasible solution by CPLEX increases. In contrast to CPLEX, Tabu search algorithm can provide good feasible solutions in a very short computational time.

Table IV-6 The results from experiments of large size problems

Problem set	Tabu time(sec)	Gap
L1	934	4.9
L2	269	-1.1
L3	217	-45.8
L4	203	-29.2
L5	705	-17.3



*This graph plots only some feasible solutions found by the Tabu search algorithm and CPLEX which are the critical points for observing the trend of the result.

Figure IV-7 Computational time and optimal gap of large size problem [part2]

In summary, the complexity of problems drastically increases when the number of resource type and the number of task that requires joint of multiple resource types increase. Tabu search algorithm performs well in all problem sizes. Many solutions in small size problem are optimal and the average gap for all problems is 2.2%. In the medium size problem, good solutions can be found in a short time comparing with CPLEX (average gap = 5.8%). For the large size problem, the proposed algorithm clearly outperforms CPLEX. Most solutions from Tabu search algorithm are better than solutions from CPLEX and all best solutions can be found quickly comparing with CPLEX.

4.5 Conclusion

The purpose of this chapter is to develop model and heuristic for the multi-period multi-site assignment problem concerning joint requirement of multiple resource types. The mathematical model and heuristic based on Tabu search algorithm is developed. The proposed Tabu search algorithm is separated into two steps (two-step Tabu search algorithm). The first step is to allocate resources to site while the second step is to assign resources to task. The experiment is done to evaluate the efficiency of the algorithm. Test problems are grouped into three sizes: small, medium and large size problems. The result shows that the developed algorithm provides good solutions in all problem sizes.

CHAPTER V
MULTI-PERIOD MULTI-SITE ASSIGNMENT PROBLEM WITH JOINT REQUIREMENT OF MULTIPLE RESOURCE TYPES AND RELOCATION OF RESOURCES (MM-JM-Re)

In this chapter the mathematical model and heuristic for multi-period multi-site assignment problem with joint requirement of multiple resource types and relocation of resource are developed. This is the complete model of the proposed problem. In this model, there are multiple resource types and all resources can be rotated. The mathematical model and heuristic is developed based on mathematical model and Tabu search algorithm in Chapter IV.

The remainder of this chapter is organized as follows. In Section5.1, the problem description is presented. The mathematical model and heuristic algorithm are described in Section 5.2 and 5.3 respectively. Then, the computational experiment is shown in Section 5.4. Finally, in Section 5.5, the conclusion is done.

5.1 Problem description

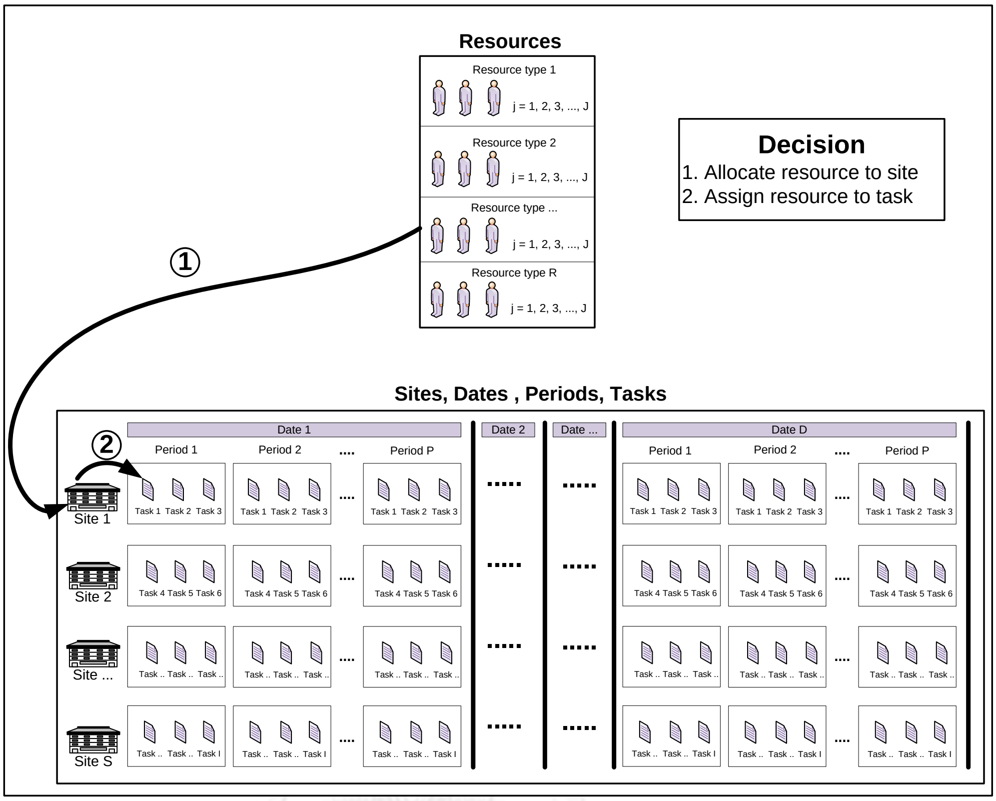


Figure V-1 Characteristics of the MM-JM-Re model [1/2]

In this model, from Fig.V-1, there are many resource types (r= 1, 2,..., R) and in each type there are many resources (j = 1, 2,..., J). There are many sites (s = 1, 2,..., S) and days (d = 1, 2,..., D). In each day, there are many periods (p = 1, 2,..., P). In each period, there are many tasks (i = 1, 2,..., I).

Resources can be rotated to any sites; however, the resource can be assigned to only one site per day and done only one task per period. An example of relocation of resource is illustrated in Fig.V-2(A). The considered resource is assigned to Site 2 in Date1, Site 3 in Date 2, Site 1 in Date 3 and Site 2 in Date 4. After resources finish their tasks on the last date, they must be returned to sites where they are assigned on the first date.

Tasks may require one or more than one resource types for operations as shown in Fig.V-2(B). Task 1 requires two resource types (Resource type 1 and Resource type 2) while Task 2 requires three resource types (Resource type 1, Resource type 2 and Resource type 3). For Task 3, only Resource type 4 is required. Only qualified resources can do tasks and task is done only when joint requirements of resources are satisfied.

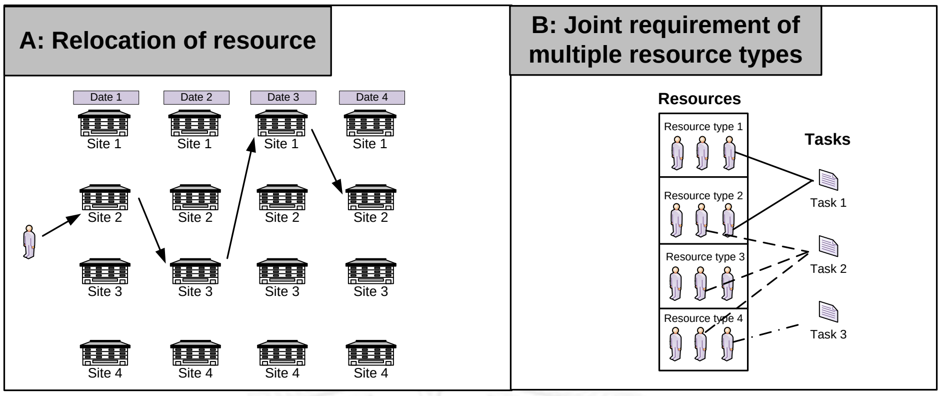


Figure V-2 Characteristics of the MM-JM-Re model [2/2]

The decision is to allocate resources to sites and assign resources to tasks to maximize total profit which is calculated from total benefit, total operation cost and total transportation cost. Total benefit is calculated from benefit from executed tasks in each period. Total operation cost is calculated from cost of assigning each resource to site and total transportation cost is calculated from cost of relocating resources to another site at the end of each day.

5.2 Mathematical model

From the problem description, a mathematical model can be written as follows.

Index

d = index for dates; $d \in \{1, 2, 3, \ldots, D\}$

i = index for tasks; $i \in \{1, 2, 3, \ldots, I\}$

j = index for resources; $j \in \{1, 2, 3, \ldots, J\}$

p = index for periods; $p \in \{1, 2, 3, \ldots, P\}$

r = index for resource types; $r \in \{1, 2, 3, \ldots, R\}$

s = index for sites; $s \in \{1, 2, 3, \ldots, S\}$

Set

I_{dps} = set of task i occurring in site s in period p in date d

Parameters

g^{r}_{jdpi} =1 if the resource j in type r is qualified to do task i in period p in date d.

= 200 otherwise. [Big M value]

b^{r}_{dpi} = 1 if task i in period p in date d requires resource type r.

= 0 otherwise.

B_{dpi} = benefit when task i in period p in date d is executed.

C^{o}_{rjds} = operation cost when resource j in type r in date d is assigned to site s.

$C^{t}_{dss'}$ = transportation cost when the resource move from site s in date d to site s' in date d+1

Decision variables

Y^{r}_{jdpi} = 1 if resource j in type r is assigned to task i in period p in date d.

= 0 otherwise.

Z^{r}_{jds} = 1 if resource j in type r is assigned to site s in date d.

= 0 otherwise.

W_{dpi} = 1 if task i in period p in date d is executed.

= 0 otherwise.

Objective function

Maximize total profit = $\sum_{d=1}^{D}\sum_{p=1}^{P}\sum_{i=1}^{I} B_{dpi}W_{dpi} - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{d=1}^{D}\sum_{s=1}^{S} C^{o}_{rjds}Z^{r}_{jds}$

$$-\sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{d=1}^{D-1}\sum_{s=1}^{S}\sum_{s'=1}^{S} C^{t}_{dss'}(Z^{r}_{jds}\times Z^{r}_{j(d+1)s'}) - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{s=1}^{S}\sum_{s'=1}^{S} C^{t}_{Dss'}(Z^{r}_{jDs}\times Z^{r}_{j1s'}) \tag{5.1}$$

Constraints

Qualification constraint:

$$\sum_{i=1}^{I} g^{r}_{jdpi} Y^{r}_{jdpi} \le 1 \;; \forall r \in \{1,\ldots,R\},\, j \in \{1,\ldots,J\},\, d \in \{1,\ldots,D\},\, p \in \{1,\ldots,P\} \tag{5.2}$$

Location constraint:

$$\sum_{s=1}^{S} Z^{r}_{jds} = 1 \;; \forall r \in \{1,\ldots,R\},\, j \in \{1,\ldots,J\},\, d \in \{1,\ldots,D\} \tag{5.3}$$

Joint requirement constraint:

$$\sum_{j=1}^{J} g^{r}_{jdpi} Y^{r}_{jdpi} = b^{r}_{dpi} W_{dpi} \;; \forall r \in \{1,\ldots,R\},\, d \in \{1,\ldots,D\},\, p \in \{1,\ldots,P\},\, i \in \{1,\ldots,I\} \tag{5.4}$$

Available task constraint:

$$Z^{r}_{jds} \ge Y^{r}_{jdpi} \;; \forall r \in \{1,\ldots,R\},\, j \in \{1,\ldots,J\},\, d \in \{1,\ldots,D\},\, p \in \{1,\ldots,P\},\, s \in \{1,\ldots,S\},\, i \in I_{dps} \tag{5.5}$$

The objective function, Eq. (5.1), maximizes the total profit, which is calculated from total benefits, total operation costs and total transportation costs. The first term in Eq. (5.1) is the sum of benefit whereas the second term is the sum of operation cost. For the third and the forth terms, they are the sum of transportation cost for relocating resources from sites to other sites and returning them back to sites on the first date respectively. Eq. (5.2) enforces that only qualified resources can do tasks and each resource is assigned to only one task per period. Eq. (5.3) enforces that each resource must be assigned to only one site per day. Eq. (5.4) states that only qualified resources can do tasks and tasks can be done when joint requirements of resources are satisfied. Each site has different tasks and resources can do only tasks in the site where they are assigned. Eq. (5.5) is used for enforcing this restriction.

<u>Quadratic transformation</u>

The above mathematical model is a quadratic integer programming problem which is not easy to be solved. Aronson [60] suggested the method to transform the quadratic term in a multi-period assignment problem to a linear term, which can be applied to this proposed model.

The quadratic term in objective function will be transformed to linear term by replacing this term with new variables and adding some additional constraints.

New Decision variables

$T^{r}_{jdss'}$ = 1 if resource id j in type r moves from site s in date d to site s' in date d+1.

= 0 otherwise.

A term of $(Z^{r}_{jds} \times Z^{r}_{j(d+1)s'})$ and $(Z^{r}_{jDs} \times Z^{r}_{j1s'})$ in objective function, Eq. (5.1), is replaced by $T^{r}_{jdss'}$ and $T^{r}_{jDss'}$. Then, the objective function can be rewritten as follow, Eq. (5.6).

$$\text{Maximize total profit} = \sum_{d=1}^{D}\sum_{p=1}^{P}\sum_{i=1}^{I} B_{dpi}W_{dpi} - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{d=1}^{D}\sum_{s=1}^{S} C^{o}_{rjds}Z^{r}_{jds} - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{d=1}^{D-1}\sum_{s=1}^{S}\sum_{s'=1}^{S} C^{t}_{dss'}T^{r}_{jdss'} - \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{s=1}^{S}\sum_{s'=1}^{S} C^{t}_{Dss'}T^{r}_{jDss'} \quad (5.6)$$

The following constraints, Eq. (5.7) and Eq. (5.8), are also added to the model.

$$Z^{r}_{jds} + Z^{r}_{j(d+1)s'} - 1 \le T^{r}_{jdss'} \; ; \forall r \in \{1,...,R\}, j \in \{1,...,J\}, d \in \{1,...,D-1\}, s \in \{1,...,S\}, s' \in \{1,...,S\} \quad (5.7)$$

$$Z^{r}_{jDs} + Z^{r}_{j0s'} - 1 \le T^{r}_{jDss'} \; ; \forall r \in \{1,...,R\}, j \in \{1,...,J\}, s \in \{1,...,S\}, s' \in \{1,...,S\} \quad (5.8)$$

From this transformation, the model is changed to a linear integer programming problem, which can be solved easier by CPLEX. However, this problem is still in the class of NP problems [34] and hard to find optimal solutions when the problems become large. Tabu search heuristic from the previous model (MM-JM-NoRe) will be modified and improved to suitable for this problem.

5.3 Tabu search heuristic for model with relocation of resources (TS-2)

Tabu search algorithm in the previous chapter is developed for the model without relocation of resource (MM-JM-NoRe). For proposed model in this chapter (MM-JM-Re), relocation of resources is allowed so the algorithm cannot be applied directly.

From the developed Tabu search algorithm in previous chapter (TS-1), the algorithm is separated into two parts: allocating resources to site (MTS) and then assigning resources to task (STS).

In this section, MTS algorithm is modified and adjusted to suitable for the proposed model in this chapter, named MMTS (modified MTS). For STS, because STS algorithm is an algorithm for assigning resource to task in each site and period after resources are moved, it can be applied directly to MM-JM-Re model.

The main process of Tabu search algorithm for model with relocation of resources (TS-2) is still the same as algorithm for model without relocation of resource (TS-1). The modifications are done in the mechanism in each process.

The process of TS-2 algorithm can be illustrated in Fig.V-3. An algorithm starts from generating an initial solution. Then, a process of finding all neighborhoods is done. Each neighborhood is a set of resources which should be moved to some sites to provide better solutions. Because getting true objective function of all neighborhoods by using Sub Tabu search algorithm (STS) takes a lot of computational time, the process of reducing the number of neighborhood by selecting only some neighborhoods with some criteria to be candidates is introduced. After candidate list is generated, Sub Tabu Search Algorithm (STS) will be done to find a solution of resource assignment of each candidate and the true objective function will be calculated. After all candidates are calculated by STS, the best candidate is selected to be a new initial solution for MMTS and the process of updating Tabu list, new solution and best known solution are done. The process is done iteratively until reaching the stopping criteria.

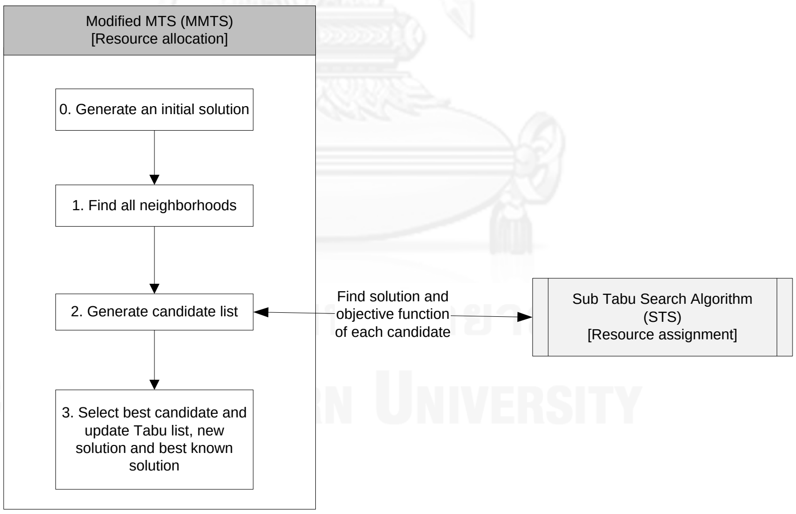


Figure V-3 Structure of Tabu search algorithm (TS-2)

5.3.2 MMTS algorithm

The details of each process of MMTS can be described as follows.

Generate an initial solution

This process is to find an initial solution. The Tabu search algorithm in the previous chapter (TS-1) is applied for find the initial solution. The details of this process can be written as follows.

1. Decompose problem: The problem is decomposed into many sub-problems by date as shown in Fig.V-4. There are D sub-problems. Each sub-problem can be seen as the problem of MM-JM-NoRe.
2. Apply TS-1 algorithm: The TS-1 algorithm is applied for finding the solution of each sub-problem.

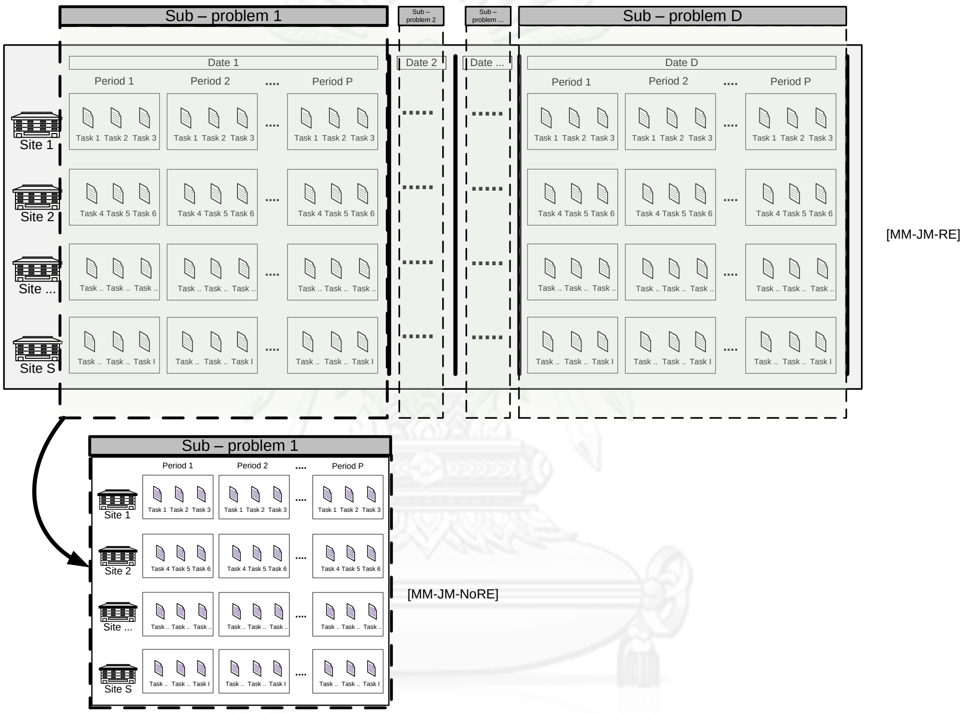


Figure V-4 Decomposing problems by date

Find all neighborhoods

The neighborhood structure is divided into two types. The neighborhood structure of each type is illustrated in Fig.V-5 and the details of each neighborhood type can be describes as follows.

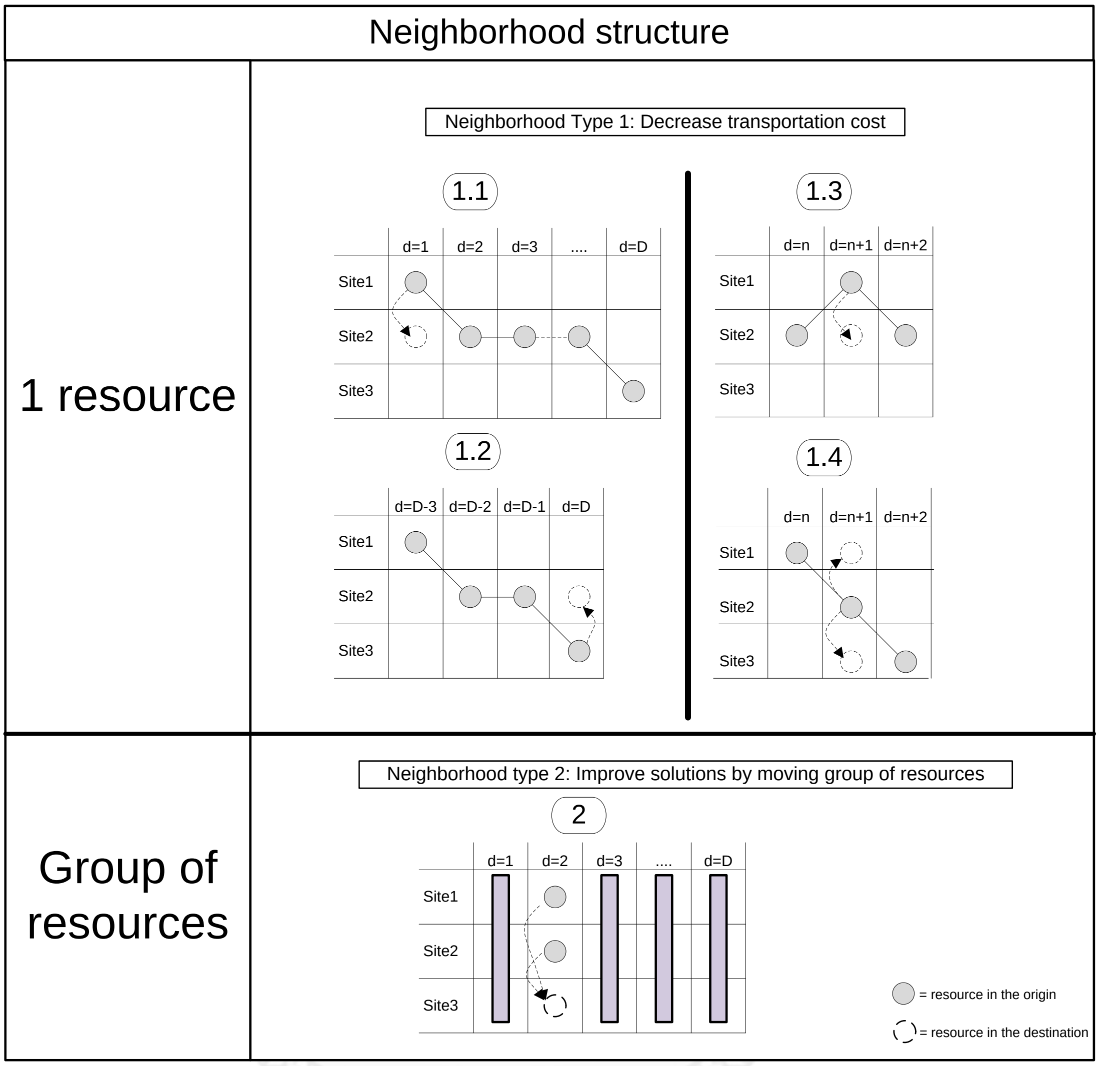


Figure V-5 Neighborhood structure

Neighborhood Type 1

- Objective of neighborhood: To decrease total transportation cost. Because the process of generating an initial solution does not concern transportation cost of each resource, total cost from moving resources from site to another site may be high. The neighborhood structure in this type tries to reduce the relocation of resource.
- Method to construct neighborhoods: The transportation cost of resources can be reduced by four methods, which are shown in Fig.V-5(1.1-1.4)
    - The first method: Move resource in the first date from the original site to the site where they are assigned in the second date (the neighborhood is to change the value of $Z^{r}_{jd_0s_{d_0}}$ from 1 to 0 and change

the value of $Z^r_{jd_0s_{d_2}}$ from 0 to 1). For example, in Fig.V-5(1.1), the resource in date d_1 is moved from site S_1 to site S_2.

- The second method: Move resource in the last date from the original site to the site where they are assigned in date d_{D-1} (the neighborhood is to change the value of $Z^r_{jd_Ds_{d_D}}$ from 1 to 0 and change the value of $Z^r_{jd_Ds_{d_{D-1}}}$ from 0 to 1). For example, in Fig.V-5(1.2), the resource in date d_D is moved from site S_3 to site S_2.
- The third method: Move resource in the date $d_{n+1}; n \in \{1,2,...,D-2\}$ from the original site to the site where they are assigned in date d_n (the neighborhood is to change the value of $Z^r_{jd_{n+1}s_{d_{n+1}}}$ from 1 to 0 and change the value of $Z^r_{jd_{n+1}s_{d_n}}$ from 0 to 1). For example, in Fig.V-5(1.3), the resource in date d_{n+1} is moved from site S_1 to site S_2.
- The forth method: Move resource in the date $d_{n+1}; n \in \{1,2,...,D-2\}$ from the original site to the site where they are assigned in date d_n and d_{n+2} (the neighborhood is to change the value of $Z^r_{jd_{n+1}s_{d_{n+1}}}$ from 1 to 0 and change the value of $Z^r_{jd_{n+1}s_{d_n}}$ and $Z^r_{jd_{n+1}s_{d_{n+2}}}$ from 0 to 1). For example, in Fig.V-5(1.4), the resource in date d_{n+1} is moved from site S_2 to site S_1 and site S_2 to site S_3.

Neighborhood Type 2

- Objective of neighborhood: To improve the quality of solutions by moving group of resources to do tasks in another site. The number of moved resource from neighborhood in type 1 is only one while in this neighborhood type there is one or more than one resource moved.
- Method to construct neighborhoods: The concept of selecting moved resources in this method is the same as concept for selecting moved resources in MTS algorithm of TS-1. The benefit and cost occurring from moving resources are traded off for selecting the best resource to be moved. The destination sites are the sites which have available tasks and selected resources are the resources that are qualified to do tasks. For example, in Fig.V-5(3), one resource from site S_1 and one resource from site S_2 are moved to site S_3.

Generate candidate list

Because there are a lot of neighborhoods in this step, we reduce the number of neighborhoods by selecting only some neighborhoods. We calculate surrogate objective function of each neighborhood, which consumes short computational time, and select only the best M neighborhoods to be the candidate in the candidate list. If there is more than one candidate generated from one task, only the candidate which has the highest surrogate objective function is considered to be in candidate list.

- Surrogate objective function = Tbg - Tbl - Toc - Ttc (5.9)
    - Tbg= Total possible benefit gain, which is the sum of highest benefits of unassigned tasks that each moved resource can do in each period in new site.
    - Tbl = Total benefit lost from moving all resource to new site.
    - Toc = Total additional operation cost from moving all resource to new site.
    - Ttc = Total additional transportation cost from moving all resource to new site.

This surrogate objective function is calculated from possible benefits that we will get and lose and additional costs that we will pay from moving resources so this surrogate objective function can accurately approximate the true objective function and is suitable for evaluating the quality of neighborhoods. Then, only good neighborhoods which have the opportunity to improve solutions will be selected to be candidates.

Tabu list

In this model, all resources from the best candidate are added to Tabu list and the resources in Tabu list are not allowed to be moved to the same sites in the same days for N iterations. This mechanism will prevent an algorithm from sending resources back to the same sites.

Stopping rule

MMTS will be run iteratively until reaching the maximum iteration W.

In conclusion, the first step of the algorithm is to generate a feasible initial solution. TS-1 is applied for finding the best solutions in each day. As a result, the solutions in each day are good. However, because of neglecting the relocation of resources in the process of generating the initial solution, total transportation cost of the initial solution may be high. The next step is to improve the solution by moving resources to new sites to reduce transportation cost or to do more tasks. The proposed neighborhood structure will generate the neighborhoods that can decrease transportation cost from reducing the relocation of resources and increase total profit from moving group of resources to do tasks in other sites.

Because there are many neighborhoods, selecting only good neighborhoods by using surrogate objective function will greatly reduce the computational time for finding good solutions. After improving process is done for many iterations, resources may be back to the same site in the same day which is the cause of being struck in a local optimum. The structure of Tabu list will prevent this problem by not allowing resources in Tabu list to move back to the same site in the same day.

The pseudo-code of the developed Tabu search algorithm (TS-2) can be written as follow.

======================Pseudo-code of TS-2 algorithm=================

Definition of parameters, variables and sets for Main Tabu Search Algorithm

$x_{initial}$ = the initial solution

x_{newsol} = the solution obtained in each iteration

$x_{bestknown}$ = the best known solution

$N(x)$ = a set of neighborhood of solution x

$C(x)$ = a set of candidate of solution x

$SurroObj(x)$ = a surrogate objective function of candidate x

$Obj(x)$ = an objective function of candidate x

$numCandidate$ = the number of neighborhood in candidate list (M)

$maxIte$ = the maximum iteration for running Tabu search algorithm (W)

$Tabulist$ = a set of resources in Tabu list

Algorithm TS2: Tabu search algorithm for multi-period multi-site assignment problem with joint requirement of multiple resource types and relocation of resource (TS-2)

Generate an initial solution: $x_{initial}$;

Set $x_{newsol} = x_{initial}$, $x_{bestknown} = x_{initial}$, $countIte = 0$;

repeat

Generate neighborhoods: $N\ (x_{newsol})$;

Let $x \in N\ (x_{newsol})$;

Set $countCandidate = 0, C\ (x_{newsol}) = \varnothing$;

while ($countCandidate < numCandidate$)

Find x which has the highest $SurroObj(x)$ and is not in $C\ (x_{newsol})$;

Add x to $C\ (x_{newsol})$;

Set $countCandidate++$;

end while

for each candidate (x) in $C\ (x_{newsol})$ **do**

Use Algorithm TS1_2 to obtain the solution: x' ;

Delete x in $C\ (x_{newsol})$ and add x' in $C\ (x_{newsol})$;

end for

Initiate a dummy solution x_{dummy} such that $Obj(x_{dummy})$ = int.MinValue;

for each candidate (x') in $C\ (x_{newsol})$ **do**

if $Obj(x') > Obj(x_{dummy})$ **then**

Set $x_{dummy} = x'$;

end if

end for

Set $x_{newsol} = x_{dummy}$;

Update $Tabulist$;

if $Obj(x_{newsol}) > Obj(x_{bestknown})$ **then**

Set $x_{bestknown} = x_{newsol}$;

end if

Set $countIte{+}{+}$;

until $countIte = maxIte$

===

5.4 Computational experiment

The developed algorithm in this chapter focuses on the decision of allocating resources to site (the decision of assigning resource to task is done by STS algorithm of TS-1). To test only the efficiency of the algorithm developed in this chapter, the number of period in all test problems is set to 1 and the decision of assigning resources to tasks is done by CPLEX.

The algorithm was coded in C# 2010 and ran on the Windows 7 Ultimate with Intel Core i5-2410M, CPU 2.30GHz and RAM 4GB. We compare our results with solutions from commercial optimization tool (ILOG CPLEX 12.6).

Test problems are generated into three different sizes. The first set of problem is a small size problem which takes short computational time. That is, CPLEX can find an optimal solution in a few second. The second set is a medium size problem which takes less than 10,000 seconds to find an optimal solution while the third set, a large size problem, takes more than 36,000 seconds (10 hours). In all problem sizes, operation cost and transportation cost are randomized uniformly between 100 to 1,000 and 50 to 500 respectively while benefit is randomized uniformly between 400 to 4,000 (proportion of transportation cost, operation cost and benefit is set to 1:2:8). A ratio of resource that can do each task is set to 0.4.

For the small size problem, the number of resource type is fixed to 2 (ratio of task that requires 1 type and 2 types is set to 25%: 75%). The number of resource is set to 2 and 4 while the number of task is set to 6 and 12. The number of date is set to 4 and 6. For the rest parameters, the number of period and site is fixed to 1 and 3 respectively. For each problem set, 10 tests are generated.

For the medium size problem, the experiment is separated into 2 parts. The number of task per site per period is varied in the first part while the number of site is varied in the second part. In the first part, the number of task per site per period is varied from 2 to 5 and the number of date is set to 6 and 10. The ratio of resource and task is set to 1:3. The number of resource type is fixed to 3 (ratio of task that requires 1 type, 2 types and 3 types is set to 20%: 20%: 60%). The number of period and site is fixed to 1 and 6 respectively. In the second part, the number of site is varied from 5 to 9. The number of resource type is fixed to 4 (ratio of task that requires 1 type, 2 types, 3 types and 4 types is set to 10%: 20%: 30%: 40%). The number of resource, period, task and date is fixed to 8, 1, 24 and 4 respectively. For each problem set, 5 tests are generated.

For the large size problem, the number of site is varied from 11 to 14 while other parameters are fixed. The number of resource and task is set to 20 and 60 respectively while the number of date and period is set to 10 and 1 respectively. The number of resource type is fixed to 3 (ratio of task that requires 1 type, 2 types and 3 types is set to 20%: 20%: 60%). For each problem set, 1 test is generated.

The details of all problem sizes are shown in Table V-1. The first nine columns are the description of tested problems, which is the size of problem, the set of problem, the number of task per site per period, the number of resource, the number of task, the number of site, the number of date, the number of period and the number of resource type, and the rest are the ratio of tasks that requires each resource type for operations.

Table V-1 Details of all problem sizes

Problem size	Problem set	Number of task/site/period	Number of resource	Number of task	Number of site	Number of date	Number of period	Number of resource types	Ratio of each resource type			
									1	2	3	4
Small problem	S1	-	2	6	3	4	1	2	0.25	0.75	-	-
	S2	-	2	6	3	6	1	2	0.25	0.75	-	-
	S3	-	4	12	3	4	1	2	0.25	0.75	-	-
	S4	-	4	12	3	6	1	2	0.25	0.75	-	-
Medium problem (Part1)	MA1.1	2	4	12	6	6	1	3	0.20	0.20	0.60	-
	MA1.2	3	6	18	6	6	1	3	0.20	0.20	0.60	-
	MA1.3	4	8	24	6	6	1	3	0.20	0.20	0.60	-
	MA1.4	5	10	30	6	6	1	3	0.20	0.20	0.60	-
	MA2.1	2	4	12	6	10	1	3	0.20	0.20	0.60	-
	MA2.2	3	6	18	6	10	1	3	0.20	0.20	0.60	-
	MA2.3	4	8	24	6	10	1	3	0.20	0.20	0.60	-
	MA2.4	5	10	30	6	10	1	3	0.20	0.20	0.60	-
Medium problem (Part2)	MB1.1	-	8	24	5	4	1	4	0.10	0.20	0.30	0.40
	MB1.2	-	8	24	6	4	1	4	0.10	0.20	0.30	0.40
	MB1.3	-	8	24	7	4	1	4	0.10	0.20	0.30	0.40
	MB1.4	-	8	24	8	4	1	4	0.10	0.20	0.30	0.40
	MB1.5	-	8	24	9	4	1	4	0.10	0.20	0.30	0.40
Large problem	L1	-	20	60	11	10	1	3	0.20	0.20	0.60	-
	L2	-	20	60	12	10	1	3	0.20	0.20	0.60	-
	L3	-	20	60	13	10	1	3	0.20	0.20	0.60	-
	L4	-	20	60	14	10	1	3	0.20	0.20	0.60	-

A parameter setting of Tabu search algorithm affects directly to the performance of the algorithm and the quality of the solution. A suitable size of Tabu list (N) will prevent an algorithm from being stuck in a local optimum. A larger number of maximum iteration (V) and candidate (M) increase the opportunity to find better solutions; however, it takes more computational time. Based on the preliminary experiments, the suitable number of maximum iteration (W), candidate (M) and Tabu list (N) are 15,000, 11, and 10 respectively. This setting may not always be a good choice for all problems. However, from our experiments which have a wide range of problem sizes and characteristics, they tend to provide good solutions.

The experiment of small size problems

In the small size problem, 4 problems are generated: S1, S2, S3 and S4. For S1 and S2, the number of resource and task are fixed to 2 and 6 whereas, for S3 and S4, the number of resource is fixed to 4 and 12. The results of the experiment are illustrated in Table V-2. The second column (#OPT by Tabu) shows the number of optimal solution found by Tabu search algorithm. The average optimal gap is shown in the third column, which is calculated from [(solution of CPLEX) – (solution of Tabu search)] *100 / (solution of CPLEX). The result shows that, for all the test problems (40 tests), 34 optimal solutions are found and the average optimal gap ranges from 0 % to 0.34 %.

Table V-2 The results from experiments of small size problems

Problem set	#OPT by Tabu (10 tests)	Average gap (%)
S1	10	0
S2	10	0
S3	9	0.02
S4	5	0.34

The experiment of medium size problems [part1]

In the medium size problem [part1], the experiment is separated into two groups and four problems per group are generated: MA1.1 to MA1.4 for the first group and MA2.1 to MA2.4 for the second group. The number of date is set to 6 for the first group and 10 for the second group. For problem in each group, all parameters are the same except the number of task and resource which is set depending on the number of task per site per period (2, 3, 4 and 5 tasks per site per period). The result of the experiment is shown in Table V-3. The second column shows the computational time of CPLEX while the third column shows time to find the best solution of Tabu search algorithm. The forth to sixth column show the minimum optimal gap, maximum optimal gap and average optimal gap. The average optimal gap and computational time are plotted in Fig.V-6. The result shows that the computational time of CPLEX extremely increases when the number of date and the number of task per site per period increase. Tabu search algorithm can find good solutions in a short time comparing with CPLEX and the quality of the solution remains good when the number of task per site per period and the number of date increase.

Table V-3 The results from experiments of medium size problems [part1]

Problem set	CPLEX time (sec)	Tabu time (sec)	Minimum gap (%)	Maximum gap (%)	Average gap (%)
MA1.1	6	2	1.03	2.51	1.85
MA1.2	18	15	3.04	3.75	3.40
MA1.3	1091	39	3.25	5.20	4.23
MA1.4	2371	63	2.55	5.19	3.83
MA2.1	22	13	2.13	4.93	3.53
MA2.2	927	41	1.87	6.25	3.91
MA2.3	3701	21	3.74	5.26	4.71
MA2.4	8096	150	3.43	5.94	4.47

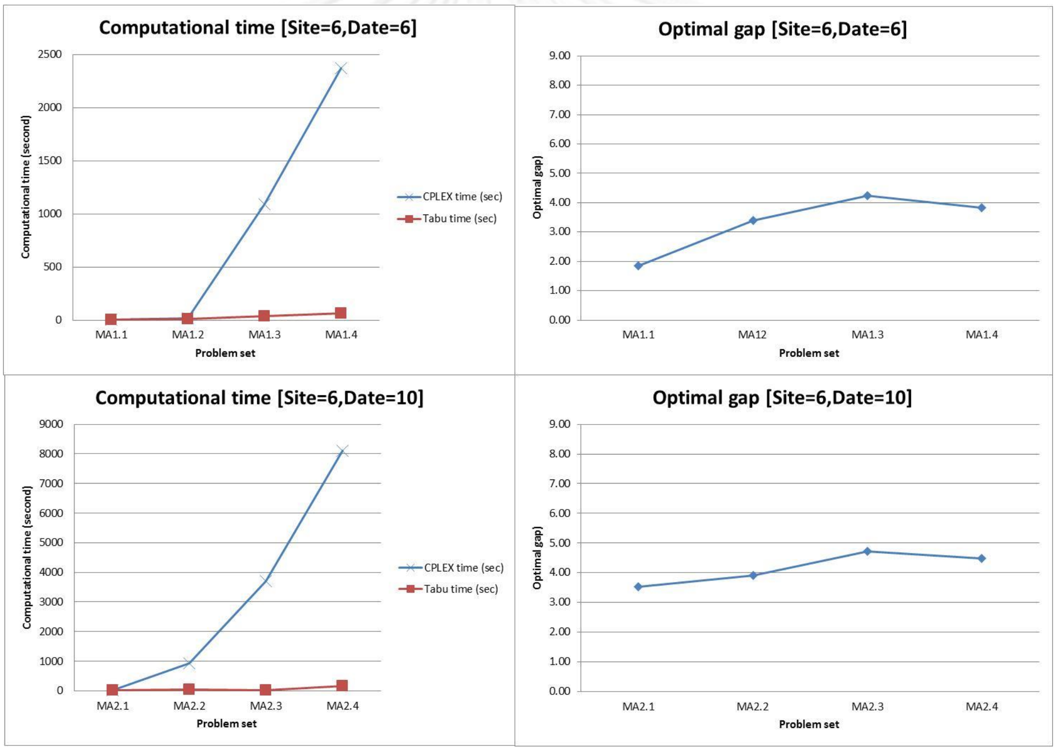


Figure V-6 Computational time and optimal gap of medium size problem [part1]

The experiment of medium size problems [part2]

In the medium size problem [part2], 5 problems are generated: MB1.1, MB1.2, MB1.3, MB1.4 and MB1.5. All parameters are fixed except the number of site which is varied from 5 to 9 (5 sites, 6 sites, 7 sites, 8 sites and 9 sites). The result of the experiment is shown in Table V-4 and the computational time and optimal gap are illustrated in Fig.V-7. The result shows that the computational time of CPLEX extremely increases when the number of site increases. The developed algorithm can find good solutions in a short time comparing with CPLEX and the optimal gap slightly increases when the number of site increases.

Table V-4 The results from experiments of medium size problems [part2]

Problem set	CPLEX time (sec)	Tabu time (sec)	Minimum gap (%)	Maximum gap (%)	Average gap (%)
MB1.1	130	91	4.32	5.99	4.81
MB1.2	674	57	3.38	5.48	4.54
MB1.3	1078	87	3.87	7.74	5.87
MB1.4	2605	51	4.38	7.17	6.21
MB1.5	5029	24	5.38	6.93	6.13

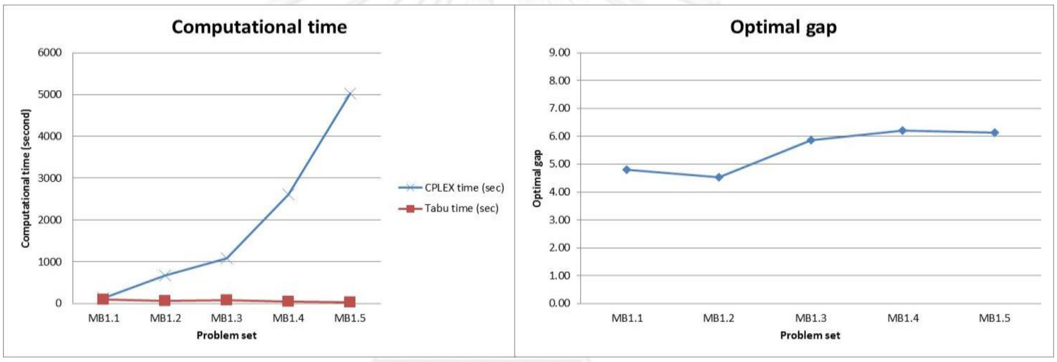


Figure V-7 Computational time and optimal gap of medium size problem [part2]

The experiment of large size problems

In the large size problem, four problems are generated: L1, L2, L3 and L4. Because finding an optimal solution for this problem set takes a lot of time, the computational time for running CPLEX is limited to 36,000 seconds (10 hours). In this experiment, the number of site is varied from 11 to 14. The result of the experiment is shown in Table V-5. The second column shows the computational time of Tabu search algorithm. The third and the fourth column show the best solutions found by Tabu search algorithm and CPLEX respectively and the fifth column is the gap between these solutions (GAP1). Because running time of CPLEX is limited, solutions from CPLEX are not optimal solutions. The sixth and the seventh column are the upper bound of the solution calculated by LP Relaxation, provided by CPLEX, and the gap between the best solution found by Tabu search algorithm and upper bound (GAP2).

The results show that although the problem size becomes large, the developed algorithm can still find good solutions in a short time (most gaps are less than 5%) and gaps between solutions from Tabu search algorithm and CPLEX do not increase comparing with medium size problems (average gaps of medium and large size problems are 4.42% and 3.77% respectively). Although these gaps are not optimal gaps, when considering Gap2 in the seventh column, which is the upper bound of the optimal gap of each test problem, the quality of the solution is still good (all gaps are less than 10%) and does not decrease when the problem size increases.

Table V-5 The results from experiments of large size problems

Problem set	Tabu time (sec)	Solution from Tabu	Solution from CPLEX	Gap1 (%)	Upper bound from CPLEX	Gap2 (%)
L1	604	549228	581321	5.52	606532.08	9.45
L2	962	551472	567423	2.81	600466.57	8.16
L3	7	561550	573719	2.12	603714.19	6.98
L4	738	551849	578501	4.61	603379.81	8.54

In summary, the complexity of problems drastically increases when the problem size increases. Tabu search algorithm performs well in all test problems. Most solutions in small size problem are optimal and the average gap for all problems is 0.09%. In the medium and large size problem, good solutions can be found in a short time comparing with CPLEX (average gap of medium size problem = 4.42% and average gap of large size problem comparing with upper bound = 8.28%).

5.5 Conclusion

The purpose of this chapter is to develop model and heuristic for the multi-period multi-site assignment problem concerning joint requirement of multiple resource types and relocation of resource. The mathematical model and heuristic based on TS-1 algorithm in Chapter IV is developed. The algorithm is separated into two steps: allocating resources to site and assigning resources to task. The algorithm for allocating resources to sites is modified from MTS algorithm of TS-1 (MMTS algorithm) while the decision of assigning resource to task can be directly applied from STS algorithm of TS-1. The experiment is done to evaluate the efficiency of the MMTS algorithm. Test problems are grouped into three sizes: small, medium and large size problems. The result shows that the developed algorithm provides good solutions in all problem sizes.

CHAPTER VI
CONCLUSION

In this study, we extend the variation of assignment problem in the dimension of task and resource by proposing the joint requirement of multiple resource types in a multi-period multi-site assignment problem. This specific characteristic is that there are multiple resource types and tasks require joint of more than one resource type for operations. This model can be found in healthcare industry, especially in clinic networks, which have many service locations, have many resource types such as physicians, nurses or medical equipments and require more than one resource type for operations. From our reviews, most multi-period multi-site assignment models consider only one resource type and if they have multiple resources types, they fail to consider joint requirement of multiple resource types.

In this research, three multi-period multi-site assignment models concerning joint requirement of multiple resource types are developed. The first model is the multi-period multi-site assignment problem with joint requirement of two resource types without relocation of resources (MM-J2-NoRe). In this model, there are only two resource types and all resources are not allowed to be rotated. For the second model, the multi-period multi-site assignment problem with joint requirement of multiple resource types without relocation of resources is developed (MM-JM-NoRe). In this model the number of resource type is not limited to two. For the last model, the multi-period multi-site assignment problem with joint requirement of multiple resource types and relocation of resources is developed (MM-JM-Re). In this model, the number of resource type is not limited to two and resources are allowed to be rotated. The mathematical model and heuristic of each problem are developed. In this chapter, all developed models and heuristics are concluded and further studies are discussed.

The remainder of this chapter is organized as follows. In Section 6.1, model description and model development is described. Mathematical models and heuristics for each model are concluded in Section 6.2. In Section 6.3, further study is summarized.

6.1 Problem description and model development

In this research, joint requirement of multiple resource types in multi-period multi-site assignment problem is considered. In this model, there are many resource types and tasks require multiple resource types for operation. Only qualified resources can do tasks. Each task provides different benefits. There are many working day and in each day working time is divided into many periods. In each period, there are many tasks and resource can do only one task in a period. There are many working sites and resources can be rotated to any sites at the end of the day. After resources finish their tasks on the last date, they must be returned to sites where they are assigned on the first date. All resources are allocated to the site with different operation cost and each resource is limited to be in one site per day. There is also transportation cost when resources are rotated. The objective is to maximize total profit which is calculated from benefit, operation cost and transportation cost.

Because our concern (joint requirement of multiple resource types) is a new extension for multi-period multi-site assignment problem, we start from developing the simple assignment model with joint requirement of two resource types (some dimensions of the proposed problem are reduced). In this model, the resource type is limited to two and resources are not allowed to be rotated. Then, the first model is extended by increasing the number of resource types to more than two (the second model) and in the third model (the complete model), the relocation of resource is included.

6.2 Mathematical models and heuristics

As described, there are three models and the mathematical model and heuristic for each model are developed. All mathematical models and heuristics are summarized as shown in Table VI-1.

Table VI-1 All developed mathematical models and heuristics

	Phase I	**Phase II**	**Phase III**
Model	MM-J2-NoRe	MM-JM-NoRe	MM-JM-Re
Core characteristics	- Two Resource types - No relocation of resource	- Multiple Resource types - No relocation of resource	- Multiple Resource types - Relocation of resource is allowed
Heuristic	Heu-1	TS-1 - MTS - STS	TS-2 - MMTS - STS

Mathematical model and heuristic for each model are developed. MM-J2-NoRe is the model for multi-period multi-site assignment problem with joint requirement of two resource types without relocation of resource. MM-JM-NoRe is the model for multi-period multi-site assignment problem with joint requirement of multiple resource types without relocation of resource. MM-JM-Re is the model for multi-period multi-site assignment problem with joint requirement of multiple resource types and relocation of resource. Heuristic for each model can be summarized as follows.

- Heu-1: This heuristic is developed for MM-J2-NoRe model. The greedy search algorithm is applied in this heuristic so this algorithm can find solutions in a very short time. However, the quality of the solution is dropped when the problem size becomes large.
- TS-1: This heuristic is developed for MM-JM-NoRe model. TS-1 is developed based on Heu-1 concept and Tabu search algorithm. The algorithm in TS-1 is divided into two steps: allocating resource to site (MTS algorithm) and assigning resource to task (STS algorithm). The quality of the solution from this algorithm is rather good (most optimal gaps of test problem is less than 10%) especially in large size problem. For large size problems, this algorithm can provide better solutions than CPLEX in a limit of time. However, in this algorithm, there are many parameters and parameter setting considerably affects to the quality of the solution. When comparing with Heu-1, TS-1 algorithm can provide higher quality solutions but takes more computational time and is more difficult to set parameters to suitable for each problem.
- TS-2: This heuristic is developed for MM-JM-Re model. The process of this algorithm is also divided into two steps: allocating resource to site and assigning resource to task. MTS algorithm of TS-1 is modified for allocating resource to site (MMTS algorithm). For the step of assigning resource to task, STS algorithm of TS-1 can be applied directly. The quality of the solution in all test problems is rather good. Most solutions in small size problem are optimal and all optimal gaps for medium and large size problems are less than 10%. Furthermore, the quality of the solution tends to remain good when the problem size increases.

6.3 Further study

The further study can be described in two aspects: improving the quality of the algorithm and including more characteristics of clinic network to the model.

1. **Improving the quality of the algorithm:** Although the developed algorithm can provide good solutions in a reasonable time, optimal gaps of some problems are rather high. Future work should find ways to improve the quality of the solution. The approach for developing heuristic of this research is to decompose the decisions of the model into two steps (allocating resources to sites and then assigning resources to tasks) and solve both steps by heuristic based on Tabu search algorithm. Another approach may be consider the whole problem without decomposition or apply the combination of meta-heuristics to find good solutions instead of using only Tabu search algorithm in both steps.
2. **Including more characteristics of clinic network to the model:** This research is motivated by the problem of health resource planning in clinic networks. The mathematical model and solution method are developed based on this problem. However, we cannot include all characteristics of clinic networks to the model. All assumptions when applying model to clinic networks are shown in Appendix A.

    Another subject of future study is to concern these assumptions and include these characteristics to the model. The assumptions and further model extensions can be summarized as follows.

    a. In the proposed model, the operation time of each task must be one period while in actual treatment operation time may be more than one period. Another subject of future study is to allow the operation time of task to more than one period.
    b. Because all service locations of the clinic network in Thailand are not too far and their resource can be moved from one location to another location within one day. In this research, the travelling is neglected. However, in some problem travelling time may be significant and cannot be neglected. Another subject of future study is to include the travelling time of the resources into the model.

c. The benefit of some clinic networks is varied depending on the individuals while in the proposed model the benefit of task is fixed. Another subject of future study is to set the benefit to be varied depending on resources.
d. Many treatments in clinic network are the continuous treatments, which require more than one time to meet the same doctor, while in the model the treatment is assumed to be one time treatment. Another subject of future study is to include continuous treatments into the model.
e. All input data in this research are deterministic but in real world problem this data is varied and stochastic. Another subject of future study is to develop model with stochastic data.
f. The number of joint requirement of resource in this research is limited to at most one resource per type while in some complex treatments they require more than one resource per type. Another subject of future study is to increase the number of joint requirement of resource to more than one resource per type.

REFERENCES

1. Foulds, L.R. and J.M. Wilson, *A variation of the generalized assignment problem arising in the New Zealand dairy industry.* Annals of Operations Research, 1997. **69**(0): p. 105-114.
2. Hajri-Gabouj, S., *A fuzzy genetic multiobjective optimization algorithm for a multilevel generalized assignment problem.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2003. **33**(2): p. 214-224.
3. Reynolds, G.H., *A Shuttle Car Assignment Problem in the Mining Industry.* Management Science, 1971. **17**(9): p. 652-655.
4. Sherali, H.D., E.K. Bish, and X. Zhu, *Airline fleet assignment concepts, models, and algorithms.* European Journal of Operational Research, 2006. **172**(1): p. 1-30.
5. Bilgin, S. and M. Azizoglu, *Operation assignment and capacity allocation problem in automated manufacturing systems.* Computers & Industrial Engineering, 2009. **56**(2): p. 662-676.
6. Corominas, A., R. Pastor, and E. Rodriguez, *Rotational allocation of tasks to multifunctional workers in a service industry.* International Journal of Production Economics, 2006. **103**(1): p. 3-9.
7. Pentico, D.W., *Assignment problems: A golden anniversary survey.* European Journal of Operational Research, 2007. **176**(2): p. 774-793.
8. Rainwater, C., J. Geunes, and H. Edwin Romeijn, *The generalized assignment problem with flexible jobs.* Discrete Applied Mathematics, 2009. **157**(1): p. 49-67.
9. Alidaee, B., H. Gao, and H. Wang, *A note on task assignment of several problems.* Computers & Industrial Engineering, 2010. **59**(4): p. 1015-1018.
10. Ravindran, A. and V. Ramaswami, *On the bottleneck assignment problem.* Journal of Optimization Theory and Applications, 1977. **21**(4): p. 451-458.
11. Gilbert, K.C. and R.B. Hofstra, *MULTIDIMENSIONAL ASSIGNMENT PROBLEMS*. 1988, Blackwell Publishing Ltd. p. 306-321.
12. Franz, L.S. and J.L. Miller, *Scheduling Medical Residents to Rotations: Solving the Large-Scale Multiperiod Staff Assignment Problem.* Operations Research, 1993. **41**(2): p. 269-279.
13. Zhang, X. and J.F. Bard, *A multi-period machine assignment problem.* European Journal of Operational Research, 2006. **170**(2): p. 398-415.
14. Gavish, B. and H. Pirkul, *Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality.* Mathematical Programming, 1985. **31**(1): p. 78-105.

15. Mazzola, J.B. and A.W. Neebe, *Resource-Constrained Assignment Scheduling.* Operations Research, 1986. **34**(4): p. 560-572.

16. Caron, G., P. Hansen, and B. Jaumard, *The Assignment Problem with Seniority and Job Priority Constraints.* Operations Research, 1999. **47**(3): p. 449-453.

17. Gutjahr, W.J. and M.S. Rauner, *An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria.* Computers & Operations Research, 2007. **34**(3): p. 642-666.

18. Costa Filho, C.F.F., et al., *Using Constraint Satisfaction Problem approach to solve human resource allocation problems in cooperative health services.* Expert Systems with Applications, 2012. **39**(1): p. 385-394.

19. Zhang, J.-H., J. Li, and Z.-P. Liu, *Multiple-resource and multiple-depot emergency response problem considering secondary disasters.* Expert Systems with Applications, 2012. **39**(12): p. 11066-11071.

20. Ozdamar, L., E. Ekinci, and B. Kucukyazici, *Emergency Logistics Planning in Natural Disasters.* Annals of Operations Research, 2004. **129**(1-4): p. 217-245.

21. Tzeng, G.-H., H.-J. Cheng, and T.D. Huang, *Multi-objective optimal planning for designing relief delivery systems.* Transportation Research Part E: Logistics and Transportation Review, 2007. **43**(6): p. 673-686.

22. Aickelin, U. and K.A. Dowsland, *An indirect Genetic Algorithm for a nurse-scheduling problem.* Computers & Operations Research, 2004. **31**(5): p. 761-778.

23. Carter, M.W. and S.D. Lapierre, *Scheduling Emergency Room Physicians.* Health Care Management Science, 2001. **4**(4): p. 347-360.

24. Aickelin, U. and K.A. Dowsland, *Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem.* Journal of Scheduling, 2000. **31**: p. 139-153.

25. Dowsland, K.A., *Nurse scheduling with tabu search and strategic oscillation.* European Journal of Operational Research, 1998. **106**(2-3): p. 393-407.

26. Burke, E., et al., *A Memetic Approach to the Nurse Rostering Problem.* Applied Intelligence, 2001. **15**(3): p. 199-214.

27. Tsai, C.-C. and S.H.A. Li, *A two-stage modeling with genetic algorithms for the nurse scheduling problem.* Expert Systems with Applications, 2009. **36**(5): p. 9506-9512.

28. Trivedi, V.M. and D.M. Warner, *A Branch and Bound Algorithm for Optimum Allocation of Float Nurses.* Management Science, 1976. **22**(9): p. 972-981.

29. Goyal, S.K. 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, 1979. **30**(5): p. 427-431.

30. Volgenant, A., *A note on the assignment problem with seniority and job priority constraints.* European Journal of Operational Research, 2004. **154**(1): p. 330-335.
31. Sheu, J.-B., *An emergency logistics distribution approach for quick response to urgent relief demand in disasters.* Transportation Research Part E: Logistics and Transportation Review, 2007. **43**(6): p. 687-709.
32. Fiedrich, F., F. Gehbauer, and U. Rickers, *Optimized resource allocation for emergency response after earthquake disasters.* Safety Science, 2000. **35**(1-3): p. 41-57.
33. Puente, J., et al., *Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms.* Computers & Industrial Engineering, 2009. **56**(4): p. 1232-1242.
34. Miller, J.L. and L.S. Franz, *A binary-rounding heuristic for multi-period variable-task-duration assignment problems.* Computers & Operations Research, 1996. **23**(8): p. 819-828.
35. Foulds, L.R. and J.M. Wilson, *On an assignment problem with side constraints.* Computers & Industrial Engineering, 1999. **37**(4): p. 847-858.
36. Toktas, B., J.W. Yen, and Z.B. Zabinsky, *Addressing capacity uncertainty in resource-constrained assignment problems.* Computers & Operations Research, 2006. **33**(3): p. 724-745.
37. Pierskalla, W.P., *Letter to the Editor--The Multidimensional Assignment Problem.* Operations Research, 1968. **16**(2): p. 422-431.
38. Mahar, S., K.M. Bretthauer, and M.A. Venkataramanan, *An algorithm for solving the multi-period online fulfillment assignment problem.* Mathematical and Computer Modelling, 2009. **50**(9-10): p. 1294-1304.
39. Romeijn, H.E. and D. Romero Morales, *A probabilistic analysis of the multi-period single-sourcing problem.* Discrete Applied Mathematics, 2001. **112**(1-3): p. 301-328.
40. Heimerl, C. and R. Kolisch, *Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets.* International Journal of Production Research. **48**(13): p. 3759 - 3781.
41. Lieshout, P.M.D. and A. Volgenant, *A branch-and-bound algorithm for the singly constrained assignment problem.* European Journal of Operational Research, 2007. **176**(1): p. 151-164.
42. Punnen, A.P. and Y.P. Aneja, *A Tabu Search Algorithm for the Resource-Constrained Assignment Problem.* The Journal of the Operational Research Society, 1995. **46**(2): p. 214-220.
43. Caunhye, A.M., X. Nie, and S. Pokharel, *Optimization models in emergency logistics: A literature review.* Socio-Economic Planning Sciences, 2012. **46**(1): p. 4-13.

44. Altay, N. and W.G. Green Iii, *OR/MS research in disaster operations management.* European Journal of Operational Research, 2006. **175**(1): p. 475-493.
45. Ernst, A.T., et al., *Staff scheduling and rostering: A review of applications, methods and models.* European Journal of Operational Research, 2004. **153**(1): p. 3-27.
46. Dowsland, K.A. and J.M. Thompson, *Solving a Nurse Scheduling Problem with Knapsacks, Networks and Tabu Search.* The Journal of the Operational Research Society, 2000. **51**(7): p. 825-833.
47. Wong, T.C., M. Xu, and K.S. Chin, *A two-stage heuristic approach for nurse scheduling problem: A case study in an emergency department.* Computers & Operations Research, 2014(0).
48. Wright, P.D. and S. Mahar, *Centralized nurse scheduling to simultaneously improve schedule cost and nurse satisfaction.* Omega, 2013. **41**(6): p. 1042-1052.
49. M'™Hallah, R. and A. Alkhabbaz, *Scheduling of nurses: A case study of a Kuwaiti health care unit.* Operations Research for Health Care, 2013. **2**(1โ€“2): p. 1-19.
50. Swangnop, S. and P. Chaovalitwongse, *A two-step tabu search heuristic for multi-period multi-site assignment problem with joint requirement of multiple resource types*, in *Manuscript submitted for publication*. 2013.
51. Daz, J.A. and E. Fernandez, *A Tabu search heuristic for the generalized assignment problem.* European Journal of Operational Research, 2001. **132**(1): p. 22-38.
52. Glover, F., *Future paths for integer programming and links to artificial intelligence.* Comput. Oper. Res., 1986. **13**(5): p. 533-549.
53. Glover, F., *Tabu search-part 1.* ORSA Journal on Computing, 1989. **1**: p. 190-206.
54. Glover, F., *Tabu search-part 2.* ORSA Journal on Computing, 1990. **2**: p. 4-32.
55. Bard, J. and L. Wan, *The task assignment problem for unrestricted movement between workstation groups.* Journal of Scheduling, 2006. **9**(4): p. 315-341.
56. Ferland, J., et al., *Generalized Assignment Type Goal Programming Problem: Application to Nurse Scheduling.* Journal of Heuristics, 2001. **7**(4): p. 391-413.
57. Burke, E., P. Causmaecker, and G. Berghe, *A Hybrid Tabu Search Algorithm for the Nurse Rostering Problem*, in *Simulated Evolution and Learning*. 1999, Springer Berlin Heidelberg. p. 187-194.
58. Li, H. and K. Womer, *Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm.* Journal of Scheduling, 2009. **12**(3): p. 281-298.
59. Chen, L., A. Langevin, and D. Riopel, *A tabu search algorithm for the relocation problem in a warehousing system.* International Journal of Production Economics, 2011. **129**(1): p. 147-156.

60. Aronson, J.E., *The multiperiod assignment problem: A multicommodity network flow model and specialized branch and bound algorithm.* European Journal of Operational Research, 1986. **23**(3): p. 367-381.
61. WHO, *Classification of health workforce statistics*.
62. WHO, *Medical device regulations: Global overview and guiding principles*. 2003.

APPENDIX

APPENDIX A
PLANNING HEALTH RESOURCES
IN CLINIC NETWORKS

In this appendix, the motivation for planning health resource in clinic network, the characteristics of the clinic network and the model formulation and assumption when applying the proposed mathematical model are presented.

Motivation

In the past, hospitals or clinics are operated individually while in recent decades (in Thailand) many clinics are formed to be a network. There are many service locations located in many regions and their health resources such as physician, nurses or medical equipments are seen as the pool of resource which can be assigned to any locations. Each service location has many customer types or treatments and each resource cannot do all treatments in the clinic so they must be rotated to many service locations to treat customers. In some upcountry, there is no physician. Physicians from Bangkok have to be rotated to those service locations. This is an example of the schedule of one physician of clinic network.

Monday: Work at service location in Ubonratchathani [by plane]

Tuesday: Work at service location in Ubonratchathani

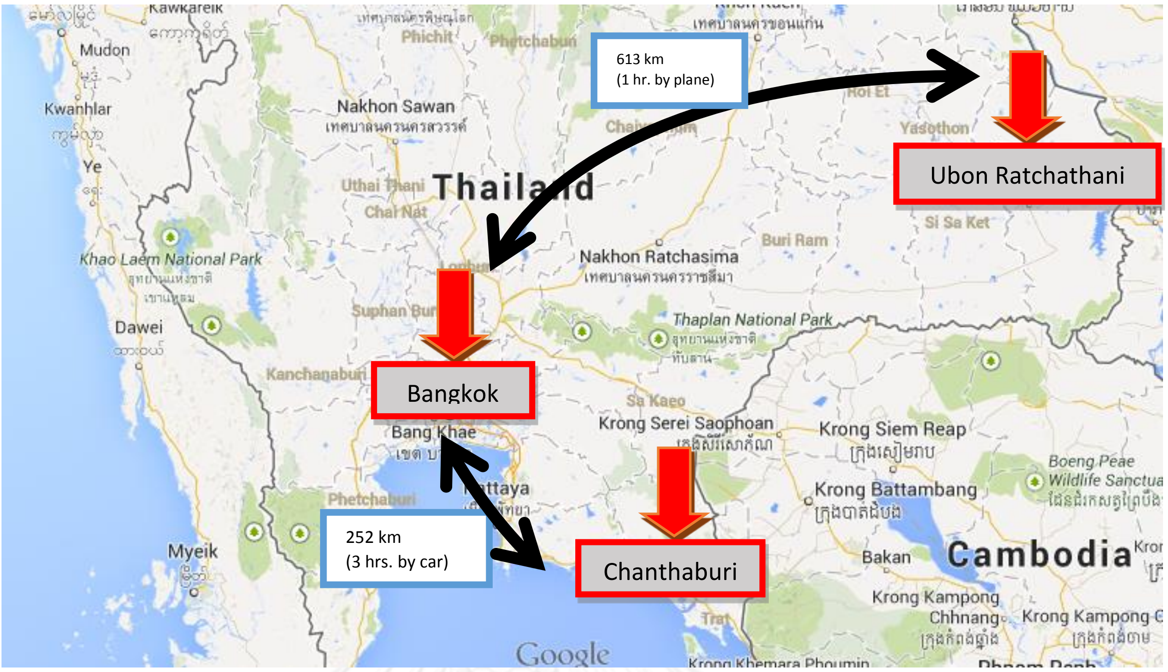
Wednesday: Day off [come back to Bangkok by plane]

Thursday: Work at service location in Chanthaburi [go by car]

Friday: Work at service location in Chanthaburi

Saturday: Work at service location in Bangkok [by car]

Sunday: Work at service location in Bangkok



It can be seen that this resource is rotated to many service locations. The clinic network will support all transportation costs and accommodation costs (operation cost) of their resources. The planner has to decide where their resource should be assigned to fulfill demands in each day to maximize total profit.

If the business size is small, such as 2-3 service locations and 5-10 resources, their resources can be planned easily but when the problem size becomes large, such as 70 service locations and 100 resources, generating a good plan to satisfy as most customers as possible while concerning related costs are not an easy task. Moreover, in the present situation, there are more new clinic networks than in the past so there are more competitions among the same clinic type. Consequently, managing on-hand health resources to optimize profit or increase the efficiency and utilization of resource is more important for this business.

<u>Characteristics of clinic network in Thailand</u>

In this section, the characteristics of clinic network in Thailand are described. Clinic network is a group of clinics which has many branches located in many areas and all of them are operated together to maximize total profit of the network. Their resources can be shared and assigned to any branches. They are mostly found in sub-specialty clinics such as dental clinics and skincare clinics. The number of branches in each network is varied from 2 to 70 branches and the number of health resource per network is varied from 20 to 100. The details of clinic network can be described in terms of resources, tasks and service location.

1. Resources

   Their main health resources are divided into three groups: physicians, nurses and medical equipments.

   I. Physicians are the person who has the clinical contact directly to the patients to promote, maintain and restore human health. The treatments that each physician can do are different.
   II. Nurses are the person who provide treatment, support and care services for people who are in need of nursing care [61]. Some nurses work in a permanent service location while some are float nurses who can be rotated to many service locations.
   III. Medical equipments are the device instruments, tools or equipments used for human beings with the specific purposes such as alleviating, monitoring, preventing or treating [62]. Most equipments are fixed in one service location while some expensive equipments are rotated.

2. Service locations

   Their service locations are located in many areas. They do not open 24/7 as the hospital. Service time of most clinic networks is around 8 to 10 hours per day (10.00 am - 20.00 pm). In each day, working hour for servicing customers is divided into many time slots or periods for treating customers and each resource can do at most one treatment per time period. They can be assigned at most one service location a day and can be moved to work in another service location at the end of the day.

3. Tasks

   The main task in clinic networks is to treat patients or customers. Some tasks are basic treatments which require only one resource such as physicians or nurses while some are specific medical treatments which require joint of resources for treating customers such as joint of physician and nurse, physician and medical equipment, nurse and medical equipment or physician, nurse and medical equipment. Benefits of each treatment are different. The planner can roughly estimate the number and type of customers in each time period.

Model formulation and assumptions

As described above, there are three types of resource: physicians, nurses and medical equipments. Some tasks require only one resource while some require joint of two or three resource types for operations. There are many service locations and working time is divided into many periods. Resources are assigned to service locations to do tasks. Operation costs for assigning resource to each service location and benefit from executing each task are different. The objective is to maximize profit which is calculated from benefit, operation cost and transportation cost. Resources must be assigned to one site and can do at most one task per period.

This problem can be formulated as an assignment model where health resources are agents, service locations are working sites and treatments are tasks. The developed assignment model, MM-JM-Re model, can be applied to this problem with the following assumptions.

- Operation time of task: Operation time of all tasks is one period.
- Travelling time: Travelling time of resources is neglected.
- Guaranteed salary: Benefit depends only on the treatment and there is no guaranteed salary for all staffs.
- Continuous treatments: Continuous treatments are out of scope of the model, so the treatment in this model is assumed as a single treatment.
- Input demand: The demands in the model are given and deterministic.
- Joint requirement condition: The number of joint requirement of resource in each resource type is at most one resource.

VITA

Mr. Siravit Swangnop was born on January 31st, 1985 in Bangkok, Thailand. He graduated from Chulalongkorn University, Thailand in academic year 2005 with Bachelor's degree and in academic year 2007 with Master's degree in Industrial Engineering. Then, he decided to continue studying in doctoral degree in Industrial Engineering at Chulalongkorn University, Thailand. During the study in master and doctoral degree, he had the opportunity to work as the research assistant in the Resource and Operation Management Research Unit (ROM) in the Faculty of Industrial Engineering at Chulalongkorn University.