TWO-SIDED ASSEMBLY LINE BALANCING MODELS AND HEURISTICS

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

UĞUR ARIKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

SEPTEMBER 2009

Approval of the thesis:

TWO-SIDED ASSEMBLY LINE BALANCING MODELS AND HEURISTICS

submitted by **UĞUR ARIKAN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Nur Evin Özdemirel ____________
Head of Department, **Industrial Engineering**

Prof. Dr. Ömer Kırca ____________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members

Prof. Dr. Sinan Kayalıgil ____________
Industrial Engineering Dept., METU

Prof. Dr. Ömer Kırca ____________
Industrial Engineering Dept., METU

Prof. Dr. Meral Azizoğlu ____________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Sedef Meral ____________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Ferda Can Çetinkaya ____________
Industrial Engineering Dept., Çankaya University

Date: 11.09.2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Uğur, ARIKAN

Signature:

ABSTRACT

TWO-SIDED ASSEMBLY LINE BALANCING

MODELS AND HEURISTICS

Arıkan, Uğur

M.S., Department of Industrial Engineering

Supervisor: Prof. Dr. Ömer Kırca

September 2009, 156 pages

This study is focused on two-sided assembly line balancing problems of type-I and type-II. This problem is encountered in production environments where a two-sided assembly line is used to produce physically large products. For type-I problems, there is a specified production target for a fixed time interval and the objective is to reach this production capacity with the minimum assembly line length used. On the other hand, type-II problem focuses on reaching the maximum production level using a fixed assembly line and workforce. Two different mathematical models for each problem type are developed to optimally solve the problems. Since the quality of the solutions by mathematical models decreases for large-sized problems due to time and memory limitations, two heuristic approaches are presented for solving large-sized type-I problem. The validity of all formulations is verified with the small-sized literature problems and the performances of the methods introduced are tested with large-sized literature problems.

Keywords: Assembly Line Balancing, Two-Sided

ÖZ

ÇİFT TARAFLI MONTAJ HATTI DENGELEME

MODELLERİ VE SEZGİSEL YÖNTEMLERİ

Arıkan, Uğur

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Ömer Kırca

Eylül 2009, 156 sayfa

Bu çalışmada tip-1 ve tip-II çift taraflı montaj bandı dengeleme problemleri üzerinde duruldu. Bu probleme fiziksel olarak büyük ürünlerin üretimi için çift taraflı montaj bandının kullanıldığı üretim ortamlarında rastlanır.Tip-I problemler için sabit bir zaman aralığı için belrlenen bir üretim hedefi bulunur ve amaç en kısa montaj bandı uzunluğu ile bu üretim kapasitesine ulaşmaktır. Diğer taraftan, tip-II problemleri sabit montaj bandı uzunluğu ve iş gücü ile en yüksek üretim seviyesine ulaşmaya odaklanır. Problemleri optimum bir şekilde çözebilmek için her tip problem için ikişer matematiksel model geliştirildi. Zaman ve hafıza sınırlamalarına bağlı olarak, büyük problemlere matematiksel modellerce bulunan sonuçların kalitesinin düşmesi nedeniyle büyük tip-I problemleri çözmek için iki sezgisel yöntem geliştirildi. Bütün formulasyonlarin geçerliliği küçük literatur problemleriyle doğrulandı ve sunulan metodların performansları büyük literatur problemleriyle test edildi.

Anahtar Kelimeler: Montaj Hattı Dengeleme, Çift Taraflı

To my family

ACKNOWLEDGEMENTS

First, I want to thank my supervisor Prof. Dr. Ömer Kırca for all kinds of support he has provided me. I also want to thank Prof. Dr. Sinan Kayalıgil, Prof. Dr. Meral Azizoğlu, Assistz Prof. Dr. Sedef Meral and Assist. Prof. Dr. Ferda Can Çetinkaya for accepting to become the jury members and for their valuable contributions.

Second, I want to thank Hüseyin Bacanak, Süleyman Arslan and Onur Aytar for sharing their expertise in computer science.

Finally, I would like to express my deepest thanks to my family and to all of my friends and to Pınar Bıçaksız. Also, for his psychological support, I want to show my greatest gratitude to my friends at METU Revolving Funds Office, especially to Avni Bal.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Assembly lines are flow-oriented production systems where the units of production performing the operations are aligned in a serial manner, referred to as *stations*. Workers and/or robots perform certain operations on the product at the stations in order to exploit a high specialization of labour and the associated learning effects (Shtub and Dar-El, 1989). The smallest individual and indivisible operations are called *tasks*. The necessary time for a task to be performed is called the *task time* or *the processing time*. Every product follows the stations along the assembly line until the raw materials turn into a final product. The operations assigned to stations are carried out on the product at each station within a specified time. This time, which is equal to the maximum of sums of processing times of the tasks in all stations, is called the *cycle time*. Production rate of the assembly line, which is the amount of final goods produced in a period of time, is directly determined by the cycle time. Assembly line balancing (ALB) problem is an assignment problem aiming to assign the tasks to the stations in order to minimize the cycle time, i.e. maximize the production rate, or minimize the line length, i.e. the workforce required.

Due to the technological and/or organizational requirements, tasks cannot be carried out in an arbitrary sequence, but they are subject to *precedence constraints*. The general input parameters of any ALB problem, precedence constraints and task times, can be summarized on a *precedence* graph. Figure 1.1 shows an example precedence graph.

Each node in the precedence graph is a task. The task times are displayed over each node. The arcs between the nodes state the precedence constraints. For instance, task 2 in Figure 1.1 has a processing time of two units and cannot be processed before task 1 is carried out.

Assembly line balancing problem is a well-studied problem in the literature. Among the types of ALB problems, this study focuses on two-sided assembly line balancing (2SALB) problem.

In chapter 2, literature review is presented in three parts. In the first part, types of assembly line balancing problems in the literature are reviewed. Second part includes the fundamental studies on general ALB problems. Final part of the chapter focuses on the studies on 2SALB.

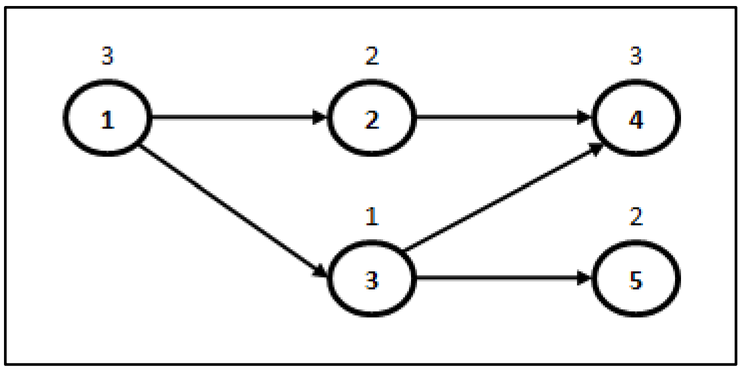

Figure 1.1 Precedence Graph

In chapter 3, mathematical models developed to solve 2SALB problems optimally are introduced. Mathematical models are developed to solve the problem for two different objectives: minimize the line length and minimize the cycle time.

- 2SALB, Type-I (minimize line length), with binary station variables
- 2SALB, Type-I (minimize line length), with integer station variables
- 2SALB, Type-II (minimize cycle time), with binary station variables

- 2SALB, Type-II (minimize cycle time), with integer station variables

Performances of the proposed mathematical models are tested with large-sized problems which are studied in the literature. Solutions found by models with binary station variables are compared with the solutions found by models with integer station variables in order to observe which performs better for 2SALB problems.

In chapter 4, two heuristic approaches are proposed for 2SALB problems with the objective of minimizing the line length:

- First heuristic approach tries to solve the entire 2SALB problem by dividing the problem into sub-problems and solving the sub-problems with a modified version of the proposed mathematical model.
- Second heuristic approach tries to find good balances by assigning tasks one by one using various selection and assignment rules.

Performances of the proposed heuristic approaches are again tested with large-sized problems that are studied in the literature.

In chapter 5, the work undertaken, results of the experiments and the possible future research directions are summarized.

CHAPTER 2

LITERATURE REVIEW

2.1 Classification of Assembly Line Balancing Problems

Many types of ALB problems are derived and studied in the literature. Among the ALB problems, the most well-known and well-studied is certainly the simple assembly line balancing (SALB) problem (Boysen, Fliedner, and Scholl, 2007). It uses many assumptions to simplify the problem without ignoring its main aspects; hence, it is regarded as the core problem of ALB.

Set of assumptions used for SALB problems are listed below (Baybars, 1986; Scholl and Becker, 2006):

1. Mass-production of one homogeneous product is carried out.
2. All tasks are processed in a predetermined mode, i.e. no alternatives for the processes exist.
3. The assembly line is a paced line with a fixed cycle time for all stations.
4. The assembly line is a serial line.
5. The processing sequence of the tasks should not violate the precedence relations.
6. The task times are deterministic.
7. There are no restrictions for the assignment of tasks except for precedence constraints.
8. A task is indivisible. Hence, it needs to be completed in a single station.
9. All stations are identical with respect to workforce, technology, etc.

A feasible line balance for a SALB problem is an assignment that does not violate the precedence relations (Boysen, Fliednerand and Scholl, 2007). SALB further assumes that the cycle times of all stations are equal to each other. Assembly lines satisfying this assumption are called *paced*. However, it is possible, inevitable in most cases, that some stations will have a sum of processing times smaller than the cycle time of the assembly line. The unproductive period of time at a station is called *idle time*. A good assembly line balance should have as few idle time as possible.

According to the objective function considered, SALB problems are further categorized in four types (Scholl and Becker, 2006):

- SALBP-I (Type-I): Minimizing the length of the assembly line for a given cycle time. This objective is equivalent to minimizing the idle times of opened stations.
- SALBP-II (Type-II): Minimizing the cycle time for a given number of stations opened.
- SALBP-E: Maximizing the *line efficiency*, *E*. This objective both considers number of stations and cycle time as a variable (Bautista and Pereira, 2006). The line efficiency is the productive fraction of the line's total operating time:
  $E = t_{sum} / (N\ CT)$, where t_{sum} is the sum of processing times of all tasks, N is the number of stations and CT is the cycle time.
- SALBP-F: This is the feasibility problem which is to establish whether or not a feasible line balance exists for a given cycle time and line length.

In the literature, the assumptions of SALB problem are relaxed and various model extensions are considered. Also, variations with respect to the objective are studied. A detailed classification of ALB problems was presented by the work of Boysen, Fliedner and Scholl (2007). The most common variations are explained below:

Mixed-Model Line: Different models of a product are produced in an arbitrarily intermixed sequence (Scholl, 1999). The task time may differ between models. Producing each model of the product requires the completion of its own set of tasks. In other words, each model has its own precedence graph.

Multi-Model Line: Multi-model line produces a sequence of batches of one model with intermediate setup operations. Hence, the ALB problem is not only a sequencing problem but also a lot sizing problem (Burns and Daganzo, 1987; Dobson and Yano, 1994).

U-Shaped Line: Instead of a straight line, the stations are arranged along a narrow "U", where both legs are closely together. This configuration allows *crossover stations*. Work pieces may revisit the same station at a later stage in the production process. This can result in better balances for cases with large number of tasks and stations (Miltenburg and Wijngaard, 1994; Scholl and Klein, 1999).

Parallel Stations: In cases when the processing times of some tasks are greater than the aimed cycle time, *parallelism* should be considered. Parallelism is the duplication of a station task group. The tasks are performed on different stations on different products simultaneously. In these problems, number of parallel stations is another decision variable to be considered.

Two-Sided Line (2SALB): These lines are necessary when assembling physically large products, such as buses and trucks. In these lines, both left and right sides of the assembly line may be used. At a time different tasks may be carried out at the sides of the stations. A two-sided assembly line is illustrated in Figure 1.2. A mated-station consists of right and left stations directly facing each other. The nature of the physically large products imposes side restrictions on the tasks. In other words, some tasks may only be performed on the left of the assembly line (L-tasks) and some tasks may only be performed on the right (R-tasks), while some tasks, without side restrictions, may be assigned to either side of the line (E-tasks). Both sides of the mated-stations are identical to each other and they are subject to the same cycle time.

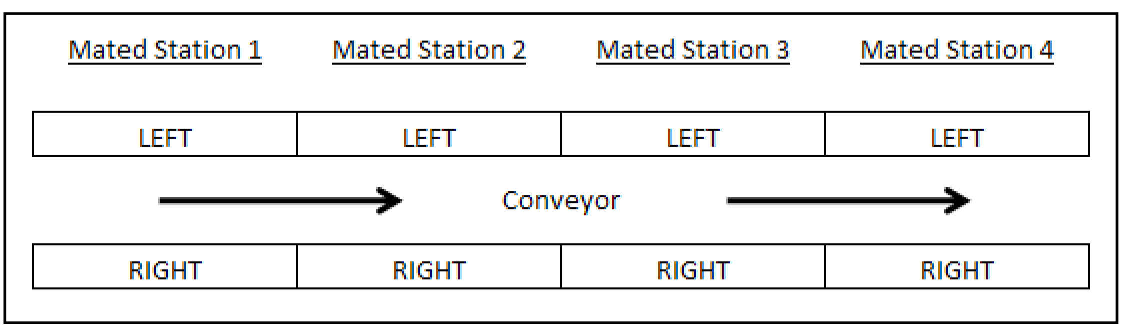

Figure 2.1 Two-Sided Assembly Line

2.2 Literature on General Assembly Line Balancing

ALB problem is first described by Bryton (1954) in his master's thesis work at Northwestern University. Salveson (1955) formulated the SALB problem as a linear programming problem incorporating all possible combinations of task assignments to stations. However, this approach can often lead to infeasible line balances since it allows task divisibility. Bowman (1960) formulated the SALB problem as an integer programming problem depicting task assignments to stations with 0-1 variables. This approach provided feasible assembly balances with indivisible tasks. Later, integer programming problem was modified by White (1961).

Integer programming (IP) formulations of SALB problem were contributed by Klein (1963); Thangavelu and Shetty (1971); Patterson and Albracht (1975); and Talbot and Patterson (1984), who formulated the problem as a general integer program without binary variables. General integer program formulation by Patterson and Albracht (1975) significantly reduced the size of the problem. Also Patterson and Albracht (1975) introduced earliest and latest station concepts. The authors presenting IP formulations (Thanganelu and Shetty 1971, Patterson and Albracht 1975, and Talbot and Patterson 1984) proposed optimal solutions using branch and bound techniques based on IP codes. On the other hand, Klein (1963) and Gutjahr and Nemhauser (1964) used shortest path techniques to solve the problem.

Helgeson and Birnie (1961) introduced Ranked Positional Weight Heuristic for solving SALB problem. Hoffman (1963) proposed a heuristic algorithm based on a precedence matrix. The heuristic generates feasible task combinations for the station under consideration using the matrix and selects the combination with minimum idle time.

Dynamic programming (DP) formulations were introduced by Jackson (1956), Held et al. (1963) and Kao and Queyranne (1982). Held and Karp (1962) and Schrage and Baker (1978) also presented DP formulations in the general context of sequencing the precedence relations. Compared to traditional DP algorithms, the labeling scheme introduced by Schrage and Baker (1978) is quite efficient according to time and memory requirements.

Jackson (1956), Hu (1961), Van Assche and Herroelen (1979), Johnson (1981) and Wee and Magazine (1981) introduced specialized branch and bound approaches to solve SALB problems.

Johnson (1988) introduced Fast Algorithm for Balancing Lines Effectively (FABLE) as a branch and bound procedure to find an optimal solution to large-sized SALB problems. FABLE is a 'laser' type, depth-first, branch-and-bound algorithm, with logic designed for very fast achievement of feasibility, ensuring a feasible solution to any line of 1000 or even more tasks. It utilizes new and existing dominance rules and bound arguments.

Hoffman (1992) introduced an exact branch and bound method for SALBP-I which guaranteed optimality. Boctor (1995) proposed a multiple-rule heuristic approach. The heuristic determines the schedulable tasks at each step and assigns the task with the highest priority. Priorities of the tasks are determined with the rules below:

1. The task having processing time equal to the remaining time of the station under consideration.
   - If there are no such tasks, the step is skipped.
   - If more than one such task exists, the task with the maximum number of immediate successors is assigned.
2. The 'severe task' having the maximum number of successors.
   - If there are no such tasks, the step is skipped.
   - If more than one such task exists, task with the largest processing time is assigned.
3. Combination of two tasks having a processing time equal to the remaining time.
   - If there are no such tasks, the step is skipped.
   - If more than one such combination exists, combination with the maximum number of immediate successors is assigned.
4. The task having the maximum number of successors.
   - If more than one such task exists, the task with the maximum number of 'severe' immediate successors is assigned.
   - If more than one such task exists, the task with the largest processing time is assigned.

'Severe task' is a task with a task time greater than or equal to one half of the cycle time.

2.3 Literature on Two-Sided Assembly Line Balancing (2SALB)

Bartholdi (1993) was first to address 2SALB problem. Bartholdi (1993) developed an interactive algorithm for balancing one-sided and two-sided assembly lines. The program uses a modified First Fit Rule (FFR). The set of schedulable tasks is created at each step. The sequence of the tasks in the set is the same as the sequence they are introduced to the program. The first task of the set is assigned. The user interaction allows modifying the sequence of the tasks in the set.

Kim et al. used genetic algorithm (GA) techniques to solve type-II 2SALB problems. The steps of the GA are presented with an encoding and decoding procedure of a possible solution to the problem. The overall framework of the GA procedure is as follows:

Step 1: Initial population is generated

Step 2: Each individual is evaluated.

Step 3: More fit individuals are selected with respect to the evaluation function value in order to pass on their good characteristics to offspring.

Step 4: A new crossover operator, called *structured one-point crossover (SOX)*, is developed. Using this operator, offspring is generated.

Step 5: A mutation operator is used to produces an offspring by introducing small changes in order to avoid a premature convergence to a local optima.

Step 6: Genetic crossover and mutation operations are followed by an adaptation procedure in order to complete the missing positions of the resulting offspring.

Lee et al. (2001) introduced two new performance measures: *work relatedness* and *work slackness*. Work relatedness measure (WR) is based on the formulation of Agrawal (1985). Work relatedness measures the interrelation of the tasks assigned to the same station. Two tasks are interrelated if one is reachable from the other on the precedence graph. Assigning interrelated tasks to a station is preferable according to this measure. Work slackness (WS) is a measure to quantify the tightness of task sequences. Use of this measure tends to put some room between two related tasks that are assigned to companion stations. In case the preceding task delays, the succeeding task will not be affected if there is sufficient slack time. This can be achieved by modifying the task sequence within a station. That is, the sequence of the tasks that do not have precedence relations may be flexibly adjusted and work slackness may be improved. The authors propose a heuristic approach using these performance measures. The heuristic approach is based on grouping the tasks.

Wu et al. (2008) proposed a branch-and-bound algorithm (B&B) to solve the balancing problem optimally. Also a non-linear mathematical model for type-I problem is introduced. Since the size of 2SALB enumeration tree is very large owing to the existence of E tasks, task assignment rules are developed and applied in order to reduce the size of the tree. Developed rules are:

Step 1: the tasks will be ranked according to its start time in the current position, the earlier it starts, the earlier it will be branched.

Step 2: ties broken, tasks with original L or R operation direction are assigned first

Step 3: ties broken, tasks with the maximal ranked positional weight are assigned first.

Step 4: ties broken, tasks with the maximal operation time are assigned first.

Step 5: ties broken, assigned randomly.

Baykasoglu and Dereli (2008) also used ant-colony optimization (ACO) technique for 2SALB problem. The objective is to minimize the number of workstations for a given cycle time. Also a secondary objective of maximizing work relatedness measured by Agrawal's formulation is used. The proposed algorithm can handle zoning constraints.

Xiaofeng et al. (2008) introduced a station-oriented enumerative algorithm depending on the concepts of earliest start time and latest start time. These values are used to develop a heuristics to assign tasks to stations as time within the cycle time of a station increases. Positions, mated stations, are considered one by one. The procedure may lead infeasible solutions violating the precedence relations. Hence, a backtracking mechanism is proposed to remove these infeasible solutions.

Kim et al. (2009) proposed a mathematical model and a genetic algorithm for 2SALB-II. This is the first mathematical model for type-II 2SALBP problem. The model uses binary station variables for each task-station-side (X_{ijk}). The mathematical model is tested on small-sized literature problems with 12, 16 and 24 tasks. Optimal solutions to the problems are found and the model is verified. However, due to time and memory requirements, MIP is not tested on large-sized problems. A neighbourhood genetic algorithm (n-GA) is developed for relatively large-sized problems. The algorithm uses a localized evolution to promote population diversity and search efficiency. Member of the population is presented by a two-dimensional grid. A single member and its surrounding eight neighbours form the subject of the genetic algorithm. The fitness of the potential solutions is measured by an evaluation function. The algorithm creates better-fit generations based on the initial population of nine members and the genetic factors formulated. The results of the genetic algorithm are tested on large-sized problems with 65, 148 and 205 tasks. The solutions of the algorithm are compared with the results obtained by one another genetic algorithm proposed by Kim et al. (2000) and the first fit rule (FFR) proposed by Bartholdi (1993).

Ozcan and Toklu (2009) proposed a mixed integer goal programming for 2SALB problem. The objective is to minimize the deviations from three specified target values in a lexicographic order:

- Number of mated-stations
- Cycle time
- Number of tasks assigned to a workstation

In the second part of the paper, fuzziness is introduced into the problem. The objective is to maximize the weighted average of fuzzy goals with a membership function.

Ozcan and Toklu (2009) introduced mathematical model and a simulated annealing algorithm for solving mixed model 2SALB problems. The proposed mathematical model aims to minimize the line length (number of mated-stations). The model also aims a secondary objective of minimizing the number of stations. The model is designed for handling positive and negative zoning constraints, fixed location constraints and synchronous task constraints. In the second part of the paper, simulated annealing algorithm is introduced. The algorithm has two objectives: weighted line efficiency and weighted smoothness index. The objectives are used to maximize line efficiency and distribute the work load evenly among the stations. These objectives provide the minimization of the number of stations.

Ozcan and Toklu (2009) proposed a tabu search algorithm for two-sided assembly line balancing. The line efficiency and the smoothness index are considered as the performance criteria. Proposed approach is tested on a set of test problems taken from literature and the computational results show that the algorithm performs well.

Simaria and Vilarinho (2009) developed a mathematical model to formally describe the two-sided mixed-model assembly line balancing problem. The objective of the model is to minimize the line length. However, the proposed model considers balancing the workloads between workstations and balancing the workloads within the workstations for different models as a secondary objective. Furthermore, an ant-colony optimization algorithm to solve type-I 2SALB problems is proposed. In the proposed procedure two ants ‘work’ simultaneously, one at each side of the line, to build a balancing solution which verifies the precedence, zoning, capacity, side and synchronism constraints of the assembly process.

CHAPTER 3

MATHEMATICAL MODELS FOR TWO-SIDED ASSEMBLY LINE BALANCING PROBLEMS

In chapter 3, two mathematical models are developed for 2SALB type-I problem and two mathematical models are developed for 2SALB type-II problem. The objective of type-I problem is to minimize the length of the assembly line which can achieve a given production rate, i.e. complete the assembly process in a given cycle time. On the other hand, the objective of type-II problem is to maximize the production rate, i.e. minimize the cycle length, for an assembly line with a fixed length. In the models for both types of problems, there is no parallel station. Hence, the cycle time has to be always greater than the maximum task time among all tasks of that model. The chapter begins with introducing the terminology, assumptions and notation used for the mathematical models. Then, the mathematical models are explained one by one. Validation of the models and parametric testing of the performance of the models are explained at the end of each type of problem.

3.1 Terminology, Assumptions and Notation

Necessary definitions on terminology, assumptions used by the models and notation are introduced prior to mathematical models.

3.1.1 Terminology

<u>Task</u>: Indivisible work element.

<u>Left (Right) Task</u>: Task that should be performed at the left (right) side of the line according to side constraints.

Either Task: Task that can be performed at either side of the line, having no side constraint.

Station: Location on assembly line on which certain tasks are done repeatedly. Each station has two mated-stations on the left and right of the line. Mated-stations can be used to process different tasks simultaneously, each having the same cycle time.

Task Time: Duration necessary to perform a task.

Cycle Time: Available time to perform all tasks assigned to a station.

Precedence Relations: Restrictions on the order of execution of the tasks due to the nature of the tasks and the products assembled.

Precedence Graph: A network based representation of precedence relations. The tasks are represented by nodes while precedence relations between tasks are represented by arcs. In addition to precedence relations, task times and side preferences of each task may be displayed in parenthesis above the task nodes in the precedence graph. An example of a precedence graph is given in the first chapter in Figure 1.1.

Predecessors of a task: Set of tasks that must be completed before the process of the considered task can start. For example, the predecessor set of task 4 in Figure 1.1 is {1, 2, 3}.

Immediate Predecessors of a task: A subset of set of predecessors of the considered task. Completion of the tasks in this set guarantees that the considered task can start, since the other tasks in the set of precedence tasks must have been completed before the tasks in the set of immediate predecessors. For example, the immediate predecessor set of task 4 in Figure 1.1 is {2, 3}.

Successors of a task: Set of tasks which cannot be started before the completion of the considered task.

Immediate Successors of a task: A subset of set of successors of the considered task. Tasks in this set can immediately start after the completion of the considered task if other predecessor tasks of these tasks are already completed.

Precedence Matrix: Precedence matrix is the matrix illustration of precedence relations. The precedence matrix of a problem with N tasks is an upper right square matrix with dimension NxN having 0's and 1's as entries:

$$P_{ij} = \begin{cases} 1 & \text{if task i immediately precedes task j} \\ 0 & \text{otherwise} \end{cases}$$

An example of a precedence matrix is given on Table 3.1 for the precedence graph given in Figure 1.1.

Table 3.1 Precedence Matrix

	1	2	3	4	5
1	-	1	1	0	0
2		-	0	1	0
3			-	1	1
4				-	0
5					-

3.1.2 Assumptions

Assumptions introduced below are applicable in all the following models and heuristics. Further assumptions will be introduced as used.

- Tasks times are deterministic.

- All stations are equally skilled with respect to the labour force and technology.
- Precedence graph is known and fixed.

3.1.3 Notation

Notation introduced below are used in all the following models and heuristics. Further notation will be introduced as needed.

Sets

L : Set of left tasks

R : Set of right tasks

E : Set of either tasks

T : Set of tasks ($T = E \cup L \cup R$)

K : Set of sides; K = {0 (right), 1 (left)}

J : Set of stations

P_i : Set of immediate predecessors of task i

P_i^* : Set of all predecessors of task i

S_i : Set of immediate successors of task i

S_i^* : Set of all successors of task i

C_i : Set of tasks that cannot be assigned to the same side with task i

$$C_i = \begin{cases} R & \text{if } i \in L \\ L & \text{if } i \in R \\ \emptyset & \text{if } i \in E \end{cases}$$

H : Set of pairs of tasks that have no precedence relations and that can be assigned to the same side of the assembly line.

$$H = \{(i, h) \mid i \in T, h \in T, h \in (T - P_i^* - S_i^* - C_i)\}$$

Indices

i, p, h : Task number $i = 1, \ldots, n(T)$; $p = 1, \ldots, n(T)$; $h = 1, \ldots, n(T)$

j : Station number $j = 1, \ldots, n(J)$

k : Side number $k = 0$ (for right tasks), 1 (for left tasks)

Parameters

t_i : Task time of task i

Dir_i : Preferred side of task i, $i \in (L \cup R)$

$$\mathrm{Dir}_i = \begin{cases} 1 & \text{if task i is a left task} \\ 0 & \text{if task i is a right task} \end{cases}$$

Decision Variables

Sta_i : Station number that task i is assigned to ($i \in T$)

Z_{ih} : Binary variable for preventing overlaps of tasks in H ($(i,h) \in H$)

This variable will be free if the tasks are assigned to different mated-stations or to the different sides of the same mated-station. Otherwise, the order of these tasks will be determined by the value of this variable.

$$Z_{ih} = \begin{cases} 1\text{, if tasks i and h are assigned to the same station and i is finished before h} \\ 0\text{, if tasks i and h are assigned to the same station and i is finished after h} \end{cases}$$

AD_i : Binary variable that states the side that the task is assigned to ($i \in T$)

$$\mathrm{AD}_i = \begin{cases} 1 & \text{if task i is assigned to the left side of a station} \\ 0 & \text{if task i is assigned to the right side of a station} \end{cases}$$

3.2 Mathematical Models for Type-I Problem

Side constraints of tasks may insert additional difficulty to scheduling tasks. While using two sides of the assembly line to process different tasks concurrently promises a better balance with a smaller cycle time or less number of stations, idle times cannot be avoided in some cases depending on side and precedence constraints. A set of tasks whose sum of processing times of left constrained tasks is much greater than the sum of processing times of right constrained tasks, or vice versa, generally causes much idle time if there is not a sufficient number of tasks without side constraints. For an extreme example, let the sum of processing times of L tasks be 100 units, the sum of processing times R tasks be 20 units and that of E tasks be 10 units. Relaxing all the other constraints, even task indivisibility constraint, the best schedule leaves 70 units of idle time. The relaxed solution is displayed in Figure 3.2. Tasks are assigned to the shaded regions of the assembly line while the remaining regions are idle.

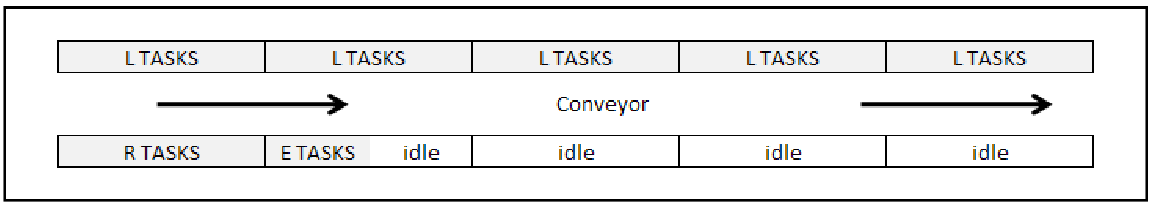

Figure 3.1 Idle Times Resulting from Side Constraints

In this chapter, two different mathematical models are developed to obtain a balance which seeks to minimize the length of the assembly line, i.e. number of mated-stations necessary for completing all the tasks in a given cycle time.

3.2.1 A Mathematical Model for Type-I Problem with Binary Station Variables

A mathematical model for Type-I 2SALB problems which uses binary station variables is developed. This model will be called as MM/Bin-I. Additional notation for MM/Bin-I is introduced in the following section.

3.2.1.1 Additional Notation

<u>Indices</u>

t : Big number index t = 1, 2, 3, 4

Parameters

CT : Cycle time

M_t : t^{th} big number

MaxN : Upper bound for the length of the line

n(J) = MaxN

Decision Variables

X_{ijk} : Binary variable for task-station-side assignment ($i \in T$, $j \in J$, $k \in K$)

$$X_{ijk} = \begin{cases} 1 & \text{if task i is assigned to side k of station j} \\ 0 & \text{otherwise} \end{cases}$$

FT_i : Finish time of task i in the station that the task is assigned to (less than or equal to the cycle time)

N : Line length, i.e. number of mated-stations

3.2.1.2 Mathematical Model

A mathematical model for solving type-I problems using binary station variables (X_{ijk}) is introduced below:

Objective

$$\min N \qquad (1)$$

Constraints

$$Sta_i = \sum_{j \in J} \sum_{k \in K} j \,.\, X_{ijk} \qquad \forall i \in T \quad (2)$$

$$X_{ij1} = 0 \qquad \forall i \in R \text{ and } \forall j \in J \quad (3)$$

$$X_{ij0} = 0 \qquad \forall i \in L \text{ and } \forall j \in J \quad (4)$$

$$\sum_{j\in J} \sum_{k\in K} X_{ijk} = 1 \qquad \forall i \in T \quad (5)$$

$$AD_i = \sum_{j\in J} \sum_{k\in K} k \, . X_{ijk} \qquad \forall i \in E \quad (6)$$

$$AD_i = Dir_i \qquad \forall i \in L \cup R \quad (7)$$

$$Sta_i \geq Sta_p \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (8)$$

$$FT_i \geq t_i \qquad \forall i \in T \quad (9)$$

$$FT_i \leq CT \qquad \forall i \in T \quad (10)$$

$$M_1 . \left(Sta_i - Sta_p\right) + FT_i \geq FT_p + t_i \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (11)$$

$$M_2 . (1 - Z_{ih}) + M_3 . (Sta_h - Sta_i) + M_4 . (AD_h - AD_i) + FT_h \geq FT_i + t_h \qquad \forall (i,h) \in H \quad (12)$$

$$M_2 . Z_{ih} + M_3 . (Sta_i - Sta_h) + M_4 . (AD_i - AD_h) + FT_i \geq FT_h + t_i \qquad \forall (i,h) \in H \quad (13)$$

$$Sta_i \leq N \qquad \forall i \in T \quad (14)$$

$$X_{ijk} \in \{0,1\} \qquad \forall i \in T, \forall j \in J \text{ and } \forall k \in K \quad (15)$$

$$AD_i \in \{0,1\} \qquad \forall i \in T \quad (16)$$

$$Z_{ih} \in \{0,1\} \qquad \forall (i,h) \in H \quad (17)$$

The objective (1) minimizes the length of the assembly line, i.e. number of mated-stations used. Constraint (2) is used for recording the station number to which each task is assigned. Constraints (3) and (4) ensure that tasks with side constraints are assigned to the proper sides. Constraint (5) ensures that every task is assigned to only one side of only one station. Constraints (6) and (7) are used for recording the side that each task is assigned to. Constraint (8) states that tasks cannot be assigned to an earlier station than the stations its predecessors are assigned to. Constraint (9) ensures that tasks cannot have a finish time smaller than its processing time. Constraint (10) restricts the finish times of the tasks with the cycle time. Constraint (11) states that tasks with precedence relations cannot be operated simultaneously when they are assigned to the same station. Constraints (12) and (13) prevent the tasks which are not connected to each other on precedence graph and are assigned to the same side of the same station from overlapping. Constraint (14) together with the objective equates the assembly line length to the maximum of the station numbers that tasks are assigned to. Constraints (15), (16) and (17) define the types of the variables.

3.2.1.3 Choice of Big Number Parameters

Four different big number parameters are used in this model. Keeping these parameters as small as possible is very important for reducing the feasible region. In addition, in order to make the mathematical model work without dividing the feasible region, relative magnitudes of these parameters need to be determined properly. Hence, the constraints using these parameters are studied in detail.

Constraint (11)

Constraint (8) ensures that Sta_i is greater than or equal to Sta_p. Hence, the left hand side of the equation cannot be negative. Then, M_1 should ensure that if these tasks are not assigned to the same station, the finish times of these tasks should be free of this constraint. CT is a good and sufficient lower bound on M_1.

$$M_1 = CT \tag{18}$$

Constraint (12) and Constraint (13)

These constraints work symmetrically. Hence, they need to be investigated together. These constraints should be passive whenever one of the following conditions is satisfied:

- The tasks are assigned to different stations
- The tasks are assigned to different sides

In addition, only one of these constraints should be active at a time, depending on the value of Z_{ih}. All possible cases are investigated for determining big number parameters.

Case I: $AD_i = AD_h$

Case I-a: $Sta_i = Sta_h$

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} + FT_i \geq FT_h + t_i$

One of the constraints will be active and one of the constraints will be passive depending on the value of Z_{ih} as expected. In order to guarantee that the constraint is passive while the other is active, the following inequality should be satisfied:

$M_2 \geq CT$

Case I-b: $Sta_i - Sta_h = d, (N-1) \geq d \geq 1$

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) – M_3.d + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} + M_3.d + FT_i \geq FT_h + t_i$

If $Z_{ih} = 1$, constraint (12) will increase FT_h and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 0 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_3 \geq CT$ (19)

$M_2 \geq M_3.MaxN$

Case I-c: $Sta_h – Sta_i = d, (N-1) \geq d \geq 1$

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) + M_3.d + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} - M_3.d + FT_i \geq FT_h + t_i$

If $Z_{ih} = 0$, constraint (13) will increase FT_i and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 1 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_3 \geq CT$

$M_2 \geq M_3.MaxN$

Case II: $AD_i - AD_h = 1$

Case II-a: $Sta_i = Sta_h$

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) - M_4 + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} + M_4 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 1$, constraint (12) will increase FT_h and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 0 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_4 \geq CT$

$M_2 \geq M_4 + CT$

Case II-b: $Sta_i - Sta_h = d$, $(N-1) \geq d > 0$

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) - M_3.d - M_4 + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} + M_3.d + M_4 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 1$, constraint (12) will increase FT_h and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 0 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_3 + M_4 \geq CT$

$M_2 \geq M_3.MaxN + M_4$ (20)

Case II-c: $Sta_h - Sta_i = d, (N\text{-}1) \geq d > 0$

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) + M_3.d - M_4 + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} - M_3.d + M_4 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 1$, constraint (12) will increase FT_h and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 0 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_2 + M_3 \geq M_4 + CT$

$M_4 \geq M_3.MaxN$ (21)

Case III: $AD_i - AD_h = -1$

Case III-a: $Sta_i = Sta_h$

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) + M_4 + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} - M_4 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 0$, constraint (13) will increase FT_i and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 1 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_4 \geq CT$

$M_2 \geq M_4 + CT$

Case III-b: Sta_i - Sta_h = d, (N-1) ≥ d > 0

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) – M_3.d + M_4 + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} + M_3.d - M_4 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 0$, constraint (13) will increase FT_i and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 1 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_4 \geq M_3.MaxN$

$M_2 + M_3 \geq M_4 + CT$

Case III-c: Sta_h – Sta_i = d, (N-1) ≥ d > 0

The constraints reduce to:

(12) $M_2.(1 - Z_{ih}) + M_3.d + M_4 + FT_h \geq FT_i + t_h$

(13) $M_2.Z_{ih} - M_3.d - M_4 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 0$, constraint (13) will increase FT_i and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 1 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_3 + M_4 \geq CT$

$M_2 \geq M_3.MaxN + M_4$

Critical inequalities are (19), (20) and (21). All other necessary inequalities are satisfied with these inequalities. Taking these three inequalities into consideration, lowest possible big number parameters are as follows:

M_2 = 2.MaxN.CT (22)

M_3 = CT (23)

M_4 = MaxN.CT (24)

Experiments show that choice of the big-M parameters is crucial in reducing the feasible region. Parameters given in equations (18), (22), (23) and (24) ensure that no feasible MIP solution is eliminated. In addition, they are the minimum possible big number parameters for the proposed mathematical model. Big number parameter calculation includes another parameter called MaxN which is the upper bound for the line length. Hence, calculation of this parameter is also very important. Upper bounds used for type-I 2SALB problems are calculated by the heuristics introduced in Chapter 4. Constraints (11), (12) and (13) are updated with the derived big number parameters and displayed below:

$$\mathrm{CT}\,.\left(\mathrm{Sta}_i - \mathrm{Sta}_p\right) + \mathrm{FT}_i \geq \mathrm{FT}_p + t_i \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (11')$$

$$2.\mathrm{maxN.CT}.(1 - Z_{ih}) + \mathrm{CT}\,.(\mathrm{Sta}_h - \mathrm{Sta}_i) + \mathrm{maxN.CT}.(\mathrm{AD}_h - \mathrm{AD}_i) + \mathrm{FT}_h \geq \mathrm{FT}_i + t_h$$

$$\forall (i,h) \in H \quad (12')$$

$$2.\mathrm{maxN.CT}.Z_{ih} + \mathrm{CT}\,.(\mathrm{Sta}_i - \mathrm{Sta}_h) + \mathrm{maxN.CT}.(\mathrm{AD}_i - \mathrm{AD}_h) + \mathrm{FT}_i \geq \mathrm{FT}_h + t_i$$

$$\forall (i,h) \in H \quad (13')$$

3.2.2 A Mathematical Model for Type-I Problem with Integer Station Variables

A mathematical model for Type-I 2SALB problems which uses only the integer station variables and excludes binary station variables is developed. This model will be called as MM/Int-I. Additional notation for MM/Int-I is introduced in the following section.

3.2.2.1 Additional Notation

Indices

t : Big number index t = 1, 2, 3, 4

Parameters

CT : Cycle time

M_t : t^{th} big number

MaxN : Upper bound for the length of the line

n(J) = MaxN

<u>Decision Variables</u>

FT_i : Finish time of task i in the assembly line (may be greater than the cycle time)

N : Line length, i.e. number of mated-stations

3.2.2.2 Mathematical Model

Developed mathematical model with integer station variables that uses only Sta_i and excludes X_{ijk} is given below:

Objective

$$\min N \tag{25}$$

Constraints

$$Sta_i \leq N \qquad \forall i \in T \tag{26}$$

$$Sta_i \geq 1 \qquad \forall i \in T \tag{27}$$

$$AD_i = Dir_i \qquad \forall i \in L \cup R \tag{28}$$

$$Sta_i \geq Sta_p \qquad \forall i \in T \text{ and } \forall p \in P_i \tag{29}$$

$$FT_i \geq (Sta_i - 1).CT + t_i \qquad \forall i \in T \tag{30}$$

$$FT_i \leq Sta_i.CT \qquad \forall i \in T \tag{31}$$

$$FT_i \geq FT_p + t_i \qquad \forall i \in T \text{ and } \forall p \in P_i \tag{32}$$

$$M_1.(1 - Z_{ih}) + M_2.(AD_h - AD_i) + FT_h \geq FT_i + t_h \qquad \forall (i,h) \in H \tag{33}$$

$$M_1.Z_{ih} + M_2.(AD_i - AD_h) + FT_i \geq FT_h + t_i \qquad \forall (i,h) \in H \tag{34}$$

$$Sta_i \in Z \qquad \forall i \in T \tag{35}$$

$$AD_i \in \{0,1\} \qquad \forall i \in T \quad (36)$$

$$Z_{ih} \in \{0,1\} \qquad \forall (i,h) \in H \quad (37)$$

The objective (25) minimizes the length of the assembly line, i.e. number of mated-stations used. Constraint (26) together with the objective equates the length of the assembly line to the maximum of the station numbers that tasks are assigned to. Constraint (27) gives the lower bound 1 for station numbers. Constraint (28) fixes the side of left and right tasks. Constraint (29) states that tasks cannot be assigned to an earlier station than the stations its predecessors are assigned to. Constraint (30) ensures that precedence relations are satisfied with respect to the task schedule. Constraints (31) and (32) ensure that FT_i lies between the starting time of the station that the task is assigned and the finishing time of that station. Constraints (33) and (34) prevent tasks with no precedence relation and which are assigned to the same side from overlapping. Constraints (35), (36) and (37) state the type of the variables.

3.2.2.3 Choice of Big Number Parameters

Two different big number parameters are used in this model. In order to keep these parameters as small as possible and make the mathematical model run properly without restricting any feasible MIP solution, the constraints that use these parameters are studied in detail.

Constraint (33) and Constraint (34)

These constraints work symmetrically. Hence, they need to be investigated together. These constraints should be passive whenever one of the following conditions is satisfied:

- The tasks are assigned to different stations
- The tasks are assigned to different sides

In addition, only one of these constraints should be active at a time, depending on the value of Z_{ih}. All possible cases are investigated for determining big number parameters.

Case I: $AD_i = AD_h$

The constraints reduce to:

(33) $M_1.(1 - Z_{ih}) + FT_h \geq FT_i + t_h$

(34) $M_1.Z_{ih} + FT_i \geq FT_h + t_i$

One of the constraints will be active and one of the constraints will be passive depending on the value of Z_{ih} as expected. Maximum value of $(FT_i – FT_h)$ and $(FT_h – FT_i)$ is $(N – 1).CT$. Hence, in order to guarantee that the constraint is passive while the other is active, the following inequality should be satisfied:

$M_1 \geq MaxN.CT$

Case II: AD_i - AD_h = 1

The constraints reduce to:

(33) $M_1.(1 - Z_{ih}) – M_2 + FT_h \geq FT_i + t_h$

(34) $M_1.Z_{ih} + M_2 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 1$, constraint (12) will increase FT_h and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 0 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_1 \geq M_2 + MaxN.CT$ (38)

$M_2 \geq MaxN.CT$ (39)

Case III: AD_i - AD_h = -1

The constraints reduce to:

(33) $M_1.(1 - Z_{ih}) + M_2 + FT_h \geq FT_i + t_h$

(34) $M_1.Z_{ih} – M_2 + FT_i \geq FT_h + t_i$

If $Z_{ih} = 0$, constraint (13) will increase FT_i and the objective function value. Hence, such a case will be eliminated by the model by forcing Z_{ih} to be equal to 1 in this case. In order to guarantee that the constraints are passive, below inequalities need to be satisfied:

$M_2 \geq MaxN.CT$

$M_1 \geq M_2 + MaxN.CT$

Critical inequalities are (38) and (39). All other necessary inequalities are satisfied with these inequalities. Taking these two inequalities into consideration, lowest possible big number parameters are as follows:

$$M_1 = 2.\mathrm{MaxN.CT} \tag{40}$$

$$M_2 = \mathrm{MaxN.CT} \tag{41}$$

Constraints (33) and (34) are updated using the derived big number parameters and displayed below:

$$2.\mathrm{MaxN.CT}.(1 - Z_{ih}) + \mathrm{MaxN.CT}.(AD_h - AD_i) + FT_h \geq FT_i + t_h \qquad \forall (i,h) \in H \quad (33')$$

$$2.\mathrm{MaxN.CT}.Z_{ih} + \mathrm{MaxN.CT}.(AD_i - AD_h) + FT_i \geq FT_h + t_i \qquad \forall (i,h) \in H \quad (34')$$

3.2.3 Computational Experiments and Comparison of the Performances of the Models

Three large-sized problems that are focused in the literature are selected for testing the performances of the mathematical models: 148-task problem (Bartholdi, 1993), 65-task problem and 205-task problem (Lee et al., 2001). The problems will be named as P148, P65 and P205 respectively and the data of the problems may be found in Appendix A. No solutions to these problems with mathematical models are available in the literature yet. Solutions to the mathematical models with binary (MM/Bin-I) and integer (MM/Int-I) station variables proposed in this study are compared. The mathematical models are developed using AMPL (a modeling language for mathematical programming) and solved using the version 10.1 of CPLEX. A cutoff after 10800 s run is used for all of the problems, that is, if the optimal solution cannot be found in 10800 seconds, current best solution is returned. AMPL code of MM/Int-I is given in Appendix E.

Two basic performance criteria are used for testing the quality of the solutions:

- <u>Line Length Gap</u>: The percentage of the gap between the best solution found in the specified time interval (if optimality is not confirmed) and the lower bound on the number of stations.

$$\text{Gap} = \frac{(\text{Solution} - \text{LB}_\text{N})}{\text{LB}_\text{N}} \text{ x } 100\%$$

where, LB_N is calculated by the formulations proposed by Wu et al. (2008):

$$S_L = \left\lceil \left(\sum_{i \in L} t_i \right) / CT \right\rceil$$

$$S_R = \left\lceil \left(\sum_{i \in R} t_i \right) / CT \right\rceil$$

$$S_E = \left\lceil \frac{\max\left(\left(\sum_{i \in E} t_i - \left((S_L + S_R).CT - \left(\sum_{i \in L} t_i + \sum_{i \in R} t_i \right) \right) \right), 0 \right)}{CT} \right\rceil$$

$$LB_N = \max(S_L + S_R) + \left\lceil \frac{\max\left((S_E - |S_L - S_R|), 0 \right)}{2} \right\rceil$$

- Solution Time: It is the time necessary to confirm optimality of the solution found. Here 'relmipgap' property of CPLEX is used to confirm optimality if a solution equal to the lower bound on the number of stations is reached. Unless the optimality is confirmed, solution time is limited to 10800 s.

The results of MM/Bin-I and MM/Int-I to P65, P148 and P205 are summarized in Table 3.2, Table 3.3 and Table 3.4 respectively.

Table 3.2 Comparison of MM/Bin-I and MM/Int-I Solutions to P65

Cycle Time	381	490	544
LB for N	7	6	5
Optimal N	7	6	5
MM/Bin-I			
Solution	8	ns	6
CPU time(s)	10800.00	10800.00	10800.00
MM/Int-I			
Solution	7*	6*	5*
CPU time(s)	4197.56	1.59	144.59

Asterisk sign, *, in the table indicates that the solution found is optimal. The abbreviation “ns” is used to indicate that no solution could be found. Complete solutions to P65 found by MM/Int-I are displayed Appendix B.

Table 3.3 Comparison of MM/Bin-I and MM/Int-I Solutions to P148

Cycle Time	204	255	306	357	408	459	510
LB for N	13	11	9	8	7	6	6
Optimal N	13	11	9	8	7	6	6
MM/Bin-I							
Solution	ns	ns	ns	ns	ns	ns	ns
CPU time(s)	10800	10800	10800	10800	10800	10800	10800
MM/Int-I							
Solution	16	14	10	10	8	7	6*
CPU time(s)	10800	10800	10800	10800	10800	10800	10800

Asterisk sign, *, in the table indicates that the solution found is optimal. The abbreviation “ns” is used to indicate that no solution could be found.

Table 3.4 Comparison of MM/Bin-I and MM/Int-I Solutions to P205

Cycle Time	1133	1510	2077
LB for N	11	8	6
Optimal N	n/a	n/a	n/a
MM/Binary			
Solution	22	ns	ns
CPU time(s)	10800	10800	10800
MM/Integer			
Solution	13	10	8
CPU time(s)	10800	10800	10800

Asterisk sign, *, in the table indicates that the solution found is optimal. The abbreviation "ns" is used to indicate that no solution could be found while "n/a" indicates that the optimal solution to the corresponding problem is not available in the literature.

From Table 3.2, it can be seen that MM/Int-I managed to find optimal solutions to the problems with different cycle times of P65 within quite reasonable times. MM/Bin-I also managed to find good solutions which use only one station more to the two of the problems. However, it failed to find a feasible solution in 10800 seconds to the other problem. The results to the problem with 148 tasks show clearly that MM/Int-I outperforms MM/Bin-I. MM/Int-I managed to find good solutions to all problems of P148 with different cycle times and furthermore, reached the optimal solution for one of the problems. On the other hand, MM/Bin-I failed to find feasible solutions to any of the problems. This is also the same case with the problems having 205 tasks, P205. MM/Bin-I found a solution to only one of the problems which is quite poor with respect to line length gap. On the other hand, MM/Int-I again managed to find good solutions to the problems with 205 tasks.

Depending on the analysis carried out with the solutions to large-sized problems, it can be claimed that mathematical model with integer station variables greatly reduces the problem size compared to mathematical model with binary station variables since MM/Int-I managed to find remarkably better solutions than MM/Bin-I. Furthermore, it may be claimed that, despite the decrease in the quality of the solution as the problem size gets larger, MM/Int-I still finds reasonable solutions in reasonable times. In order to analyze the quality of the solutions line length gap is used. Figure 3.2 displays the quality of the solutions and the trend of the quality as the problem size increases.

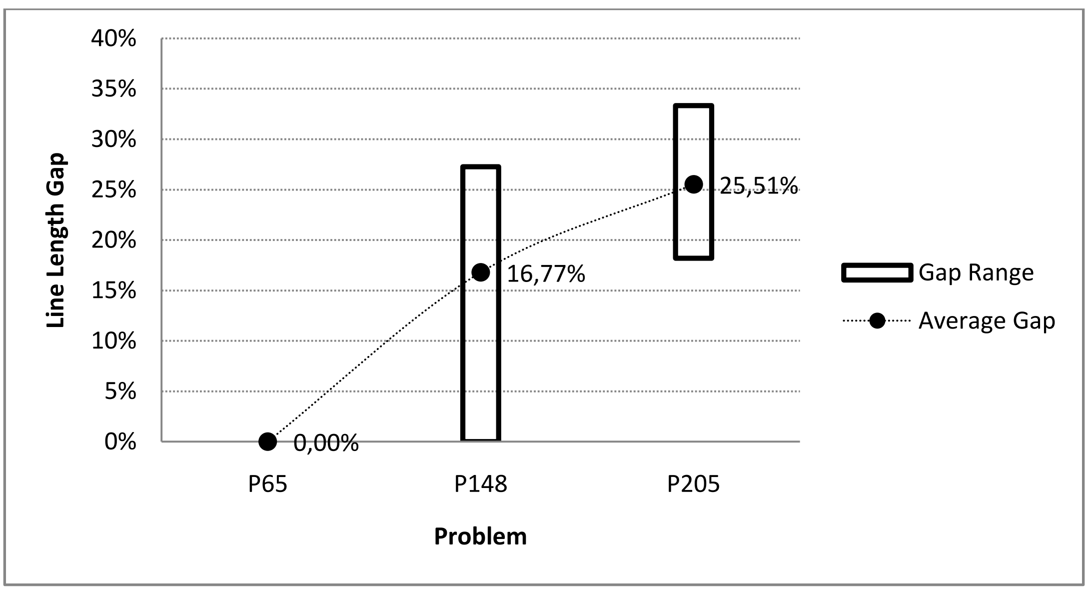

Figure 3.2 Line Length Gap Performance of MM/Int-I to Large-Sized Problems

The decrease in the quality of the solutions as the problem size gets larger may clearly be observed by Figure 3.2.

3.3 Mathematical Models for Type-II Problem

Two different mathematical models are developed to obtain a balancing which seeks to minimize the cycle time, i.e. maximize the production rate, for a fixed assembly line.

3.3.1 A Mathematical Model for Type-II Problem with Binary Station Variables

A mathematical model for Type-II 2SALB problem which uses binary station variables is developed. This mathematical model will be called as MM/Bin-II.

3.3.1.1 Additional Notation

Indices

t : Big number index $t = 1, 2, 3, 4$

Parameters

N : Number of stations $N = n(J)$

M_t : t^{th} big number

MaxCT : Upper bound for the cycle time

Decision Variables

FT_i : Finish time of task i in the station that the task is assigned to (smaller than or equal to the cycle time)

CT : Cycle time

3.3.1.2 Mathematical Model

Developed mathematical model for solving type-II problems using binary station variables (X_{ijk}) is introduced below:

Objective

$$\min \mathrm{CT} \tag{42}$$

Constraints

$$\mathrm{Sta_i} = \sum_{j\in J} \sum_{k\in K} j\,.\,X_{ijk} \qquad \forall i \in T \quad (43)$$

$$X_{ij1} = 0 \qquad \forall i \in R \text{ and } \forall j \in J \quad (44)$$

$$X_{ij1} = 0 \qquad \forall i \in L \text{ and } \forall j \in J \quad (45)$$

$$\sum_{j\in J} \sum_{k\in K} X_{ijk} = 1 \qquad \forall i \in T \quad (46)$$

$$\mathrm{AD_i} = \sum_{j\in J} \sum_{k\in K} k\,.\,X_{ijk} \qquad \forall i \in E \quad (47)$$

$$\mathrm{AD_i} = \mathrm{Dir_i} \qquad \forall i \in L \cup R \quad (48)$$

$$\mathrm{Sta_i} \geq \mathrm{Sta_p} \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (49)$$

$$\mathrm{FT_i} \geq t_i \qquad \forall i \in T \quad (50)$$

$$\mathrm{FT_i} \leq \mathrm{CT} \qquad \forall i \in T \quad (51)$$

$$M_1.\left(\mathrm{Sta_i} - \mathrm{Sta_p}\right) + \mathrm{FT_i} \geq \mathrm{FT_p} + t_i \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (52)$$

$$M_2.(1 - Z_{ih}) + M_3.(\mathrm{Sta_h} - \mathrm{Sta_i}) + M_4.(\mathrm{AD_h} - \mathrm{AD_i}) + \mathrm{FT_h} \geq \mathrm{FT_i} + t_h \qquad \forall (i,h) \in H \quad (53)$$

$$M_2.Z_{ih} + M_3.(\mathrm{Sta_i} - \mathrm{Sta_h}) + M_4.(\mathrm{AD_i} - \mathrm{AD_h}) + \mathrm{FT_i} \geq \mathrm{FT_h} + t_i \qquad \forall (i,h) \in H \quad (54)$$

$$X_{ijk} \in \{0,1\} \qquad \forall i \in T, \forall j \in J \text{ and } \forall k \in K \quad (55)$$

$$\mathrm{AD_i} \in \{0,1\} \qquad \forall i \in T \quad (56)$$

$$Z_{ih} \in \{0,1\} \qquad \forall (i,h) \in H \quad (57)$$

The objective (42) minimizes the cycle length. Constraint (43) is used for recording the station number to which each task is assigned. Constraints (44) and (45) ensure that tasks with side constraints are assigned to the proper sides. Constraint (46) ensures that every task is assigned to one side of one station. Constraints (47) and (48) are used for recording the side that each task is assigned to. Constraint (49) states that tasks cannot be assigned to an earlier station than the stations its predecessors are assigned to. Constraint (50) ensures that tasks cannot have a finish time smaller than its processing time. Constraint (51) equates the cycle time to the maximum of finish times of all tasks. Constraint (52) states that tasks with precedence relations cannot be operated simultaneously when they are assigned to the same station. Constraints (53) and (54) prevent the tasks that are assigned to the same station and the same side from overlapping. Constraints (55), (56) and (57) state the types of the variables.

3.3.1.3 Choice of Big Number Parameters

Four different big number parameters are used in constraints (52), (53) and (54). Since keeping these parameters as small as possible is important for reducing the feasible region and determining the relative magnitudes of these parameters is necessary for the model to run properly, these constraints are studied in detail (see section 3.2.1.3 for a similar study). Derived big number parameters are given below:

$$\mathrm{M_1 = MaxCT} \tag{58}$$

$$\mathrm{M_2 = 2.N.MaxCT} \tag{59}$$

$$\mathrm{M_3 = MaxCT} \tag{60}$$

$$\mathrm{M_4 = N.MaxCT} \tag{61}$$

Using derived big number parameters, updated constraints are as follows:

$$\mathrm{MaxCT.}\left(\mathrm{Sta_i - Sta_p}\right) + \mathrm{FT_i \geq FT_p + t_i} \qquad \forall \mathrm{i} \in \mathrm{T} \text{ and } \forall \mathrm{p} \in \mathrm{P_i} \quad (52')$$

$$\mathrm{2.N.MaxCT.(1 - Z_{ih}) + MaxCT.(Sta_h - Sta_i) + N.MaxCT.(AD_h - AD_i) + FT_h \geq FT_i + t_h}$$

$$\forall \mathrm{(i, h)} \in \mathrm{H} \quad (53')$$

$$\mathrm{2.N.MaxCT.Z_{ih} + MaxCT.(Sta_i - Sta_h) + N.MaxCT.(AD_i - AD_h) + FT_i \geq FT_h + t_i}$$

$$\forall \mathrm{(i, h)} \in \mathrm{H} \quad (54')$$

3.3.2 A Mathematical Model for Type-II Problem with Integer Station Variables

A mathematical model for solving type-II problems using integer station variables (excluding X_{ijk} and using only Sta_i instead) is developed. This mathematical model will be called as MM/Int-II.

3.3.2.1 Additional Notation

Indices

t : Big number index t = 1, 2, 3, 4

Parameters

N : Number of stations

M_t : t^{th} big number

MaxCT : Upper bound for the cycle time

Decision Variables

FT_i : Finish time of task i in station that the task is assigned to (smaller than or equal to the cycle time)

CT : Cycle time

3.3.2.2 Mathematical Model

Developed mathematical model for solving type-II problems with integer station variables is given below:

Objective

$$\min \mathrm{CT} \qquad (62)$$

Constraints

$$\mathrm{Sta_i} \leq \mathrm{N} \qquad \forall \mathrm{i} \in \mathrm{T} \quad (63)$$

$$\mathrm{Sta_i} \geq 1 \qquad \forall \mathrm{i} \in \mathrm{T} \quad (64)$$

$$\mathrm{AD_i} = \mathrm{Dir_i} \qquad \forall \mathrm{i} \in \mathrm{L} \cup \mathrm{R} \quad (65)$$

$$\mathrm{Sta_i} \geq \mathrm{Sta_p} \qquad \forall \mathrm{i} \in \mathrm{T} \text{ and } \forall \mathrm{p} \in \mathrm{P_i} \quad (66)$$

$$\mathrm{FT_i} \geq \mathrm{t_i} \qquad \forall \mathrm{i} \in \mathrm{T} \quad (67)$$

$$\mathrm{FT_i} \leq \mathrm{CT} \qquad \forall \mathrm{i} \in \mathrm{T} \quad (68)$$

$$\mathrm{M_1}.\left(\mathrm{Sta_i} - \mathrm{Sta_p}\right) + \mathrm{FT_i} \geq \mathrm{FT_p} + \mathrm{t_i} \qquad \forall \mathrm{i} \in \mathrm{T} \text{ and } \forall \mathrm{p} \in \mathrm{P_i} \quad (69)$$

$$\mathrm{M_2}.(1 - \mathrm{Z_{ih}}) + \mathrm{M_3}.(\mathrm{Sta_h} - \mathrm{Sta_i}) + \mathrm{M_4}.(\mathrm{AD_h} - \mathrm{AD_i}) + \mathrm{FT_h} \geq \mathrm{FT_i} + \mathrm{t_h} \qquad \forall (\mathrm{i},\mathrm{h}) \in \mathrm{H} \quad (70)$$

$$\mathrm{M_2}.\mathrm{Z_{ih}} + \mathrm{M_3}.(\mathrm{Sta_i} - \mathrm{Sta_h}) + \mathrm{M_4}.(\mathrm{AD_i} - \mathrm{AD_h}) + \mathrm{FT_i} \geq \mathrm{FT_h} + \mathrm{t_i} \qquad \forall (\mathrm{i},\mathrm{h}) \in \mathrm{H} \quad (71)$$

$$\mathrm{Sta_i} \in \mathrm{Z} \qquad \forall \mathrm{i} \in \mathrm{T} \quad (72)$$

$$\mathrm{AD_i} \in \{0,1\} \qquad \forall \mathrm{i} \in \mathrm{T} \quad (73)$$

$$\mathrm{Z_{ih}} \in \{0,1\} \qquad \forall (\mathrm{i},\mathrm{h}) \in \mathrm{H} \quad (74)$$

The objective (62) minimizes the cycle length. Constraint (63) states the upper bound on the station number variable of tasks. Constraint (64) gives the lower bound 1 for station numbers. Constraint (65) satisfies the side constraints of left and right tasks. Constraint (66) states that tasks cannot be assigned to an earlier station than the stations its predecessors are assigned to. Constraints (67) and (68) ensure that FT_i lies between the task time and the cycle time. Constraint (69) states that tasks with precedence relations cannot be operated simultaneously if they are assigned to the same station. Constraints (70) and (71) prevent tasks with no precedence relation and which are assigned to the same station and to the same side from overlapping. Constraints (72), (73) and (74) state the types of the variables.

3.3.2.3 Choice of Big Number Parameters

Four different big number parameters are used in constraints (69), (70) and (71). Again these constraints are studied in detail (see section 3.2.2.3 for a similar study).

Developed big number parameters are given below:

$$\mathrm{M_1} = \mathrm{MaxCT} \quad (75)$$

$$\mathrm{M_2} = 2.\mathrm{N}.\mathrm{MaxCT} \quad (76)$$

$$M_3 = \text{MaxCT} \quad (77)$$

$$M_4 = \text{N.MaxCT} \quad (78)$$

Updated constraints with the derived big number parameters are given below:

Objective

$$\text{MaxCT}.\left(\text{Sta}_i - \text{Sta}_p\right) + \text{FT}_i \geq \text{FT}_p + t_i \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (69')$$

$$2.\text{N}.\text{MaxCT}.(1 - Z_{ih}) + \text{MaxCT}.(\text{Sta}_h - \text{Sta}_i) + N.\text{MaxCT}.(\text{AD}_h - \text{AD}_i) + \text{FT}_h \geq \text{FT}_i + t_h$$

$$\forall (i,h) \in H \quad (70')$$

$$2.\text{N}.\text{MaxCT}.Z_{ih} + \text{MaxCT}.(\text{Sta}_i - \text{Sta}_h) + N.\text{MaxCT}.(\text{AD}_i - \text{AD}_h) + \text{FT}_i \geq \text{FT}_h + t_i$$

$$\forall (i,h) \in H \quad (71')$$

3.3.4 Performance Comparison of Models with Large-Sized Literature Problems

Three large-sized problems that are focused in the literature are selected for testing the performances of the mathematical models: 148-task problem (Bartholdi, 1993), 65-task problem and 205-task problem (Lee et al., 2001). The problems will be named as P148, P65 and P205 respectively and the data of the problems may be found in Appendix A. No solutions to these problems with mathematical models are available in the literature yet. Solutions to the mathematical models with binary (MM/Bin-II) and integer (MM/Int-II) station variables proposed in this study are compared. The mathematical models are developed using AMPL (a modeling language for mathematical programming) and solved using the version 10.1 of CPLEX. A cutoff after 10800 s run is used for all of the problems, that is, if the optimal solution cannot be found in 10800 seconds, current best solution is returned. AMPL code of MM/Int-II is given in Appendix E.

Two basic performance criteria are used for comparing the results:

Cycle Time: Best solution if optimality cannot be confirmed.

Solution Time: It is the required time to find the optimal solution. If optimality cannot be confirmed solution time is 10800 s since a cutoff after 10800 s run is applied.

The solutions to P65, P148 and P205 are summarized in Table 3.5, Table 3.6 and Table 3.7 respectively.

Table 3.5 Comparison of MM/Bin-II and MM/Int-II Solutions to P65

No.of stations	4	5	6	7	8
LB for CT	638	510	425	365	319
Optimal CT	n/a	n/a	n/a	n/a	n/a
MM/Bin-II					
Solution	649	525	439	388	330
CPU time(s)	10800	10800	10800	10800	10800
MM/Int-II					
Solution	652	537	443	374	336
CPU time(s)	10800	10800	10800	10800	10800

Abbreviation "n/a" indicates that the optimal solution for the corresponding problem is not available in the literature. Complete solutions for P65 found by MM/Int-II are displayed in Appendix C.

Table 3.6 Comparison of MM/Bin-II and MM/Int-II Solutions to P148

No.of stations	4	5	6
LB for CT	2919	2335	1946
Optimal CT	n/a	n/a	n/a
MM/Bin-II			
Solution	ns	4672	4552
CPU time(s)	10800	10800	10800
MM/Int-II			
Solution	3124	2749	2335
CPU time(s)	10800	10800	10800

Abbreviation “n/a” used indicates that the optimal solution for the corresponding problem is not available in the literature and abbreviation “ns” indicates that no solution is found.

Table 3.7 Comparison of MM/Bin-II and MM/Int-II Solutions to P205

No.of stations	4	5	6
LB for CT	641	513	427
Optimal CT	n/a	n/a	n/a
MM/Bin-II			
Solution	697	658	495
CPU time(s)	10800	10800	10800
MM/Int-II			
Solution	700	561	479
CPU time(s)	10800	10800	10800

Abbreviation “n/a” used indicates that the optimal solution for the corresponding problem is not available in the literature.

These solutions are the first mathematical model solutions to these large-sized problems. However, they fail to be the best results in the literature. Optimal solutions to these problems are not known yet. However, to the best of my knowledge, minimum cycle times for these problems up to now are found by the neighbourhood genetic algorithm (n-GA) proposed by Kim et al. (2009).

From Table 3.5, it can be seen that both of the proposed mathematical models find good solutions within 10800 seconds. Since the solution times are equal and the solutions are very close, it is very difficult to select the model with superior performance. However, the solutions to P148 and P205 summarized in Table 3.6 and Table 3.7 clearly show that mathematical model with integer station variables finds better solutions than the model with binary station variables. Also, it can be observed that as the number of mated-stations increases, deviation of the solutions from the lower increases within the specified time limit.

For better understanding the performances of MM/Int-II for large-sized problems, cycle length gap is used. This measure is evaluated by the below formula:

$$\text{Gap}_{\text{CT}} = \left\lceil \frac{\text{CT} - \text{LB}_{\text{CT}}}{\text{LB}_{\text{CT}}} \right\rceil \text{x } 100\%$$

Cycle length gap ranges and average cycle length gaps (average of the gaps of the solutions found to the same problem with different number of stations) of MM/Int-II for P65, P148 and P205 are displayed in Figure 3.3.

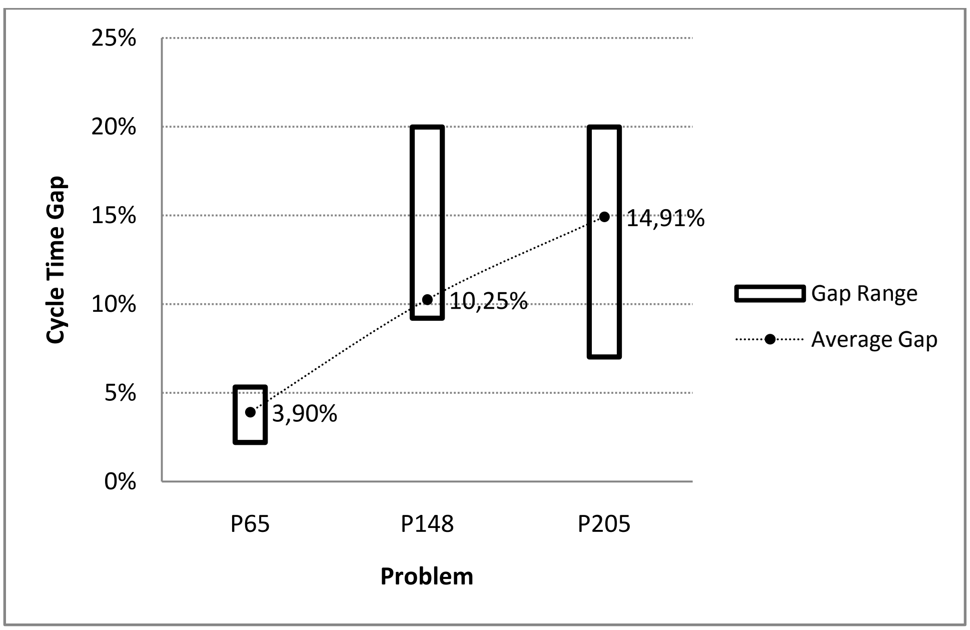

Figure 3.3 Cycle Time Gap Performance of MM/Int-II for Large-Sized Problems

It may clearly seen from Figure 3.4 that MM/Int-II finds solutions close to the lower bound for the problem with 65 tasks. However, the gap between the lower bound and the solution found by the proposed mathematical model increases as the problem size gets larger. It can be claimed that heuristic approaches may be preferred for a problem whose size is as large as that of P148 and P205.

CHAPTER 4

HEURISTIC APPROACHES

The development in the computer technology promises better solutions to large-sized problems found by the mathematical models. However, the performance of the models is limited with solution time and memory. This leads the authors to develop faster algorithms that find good solutions to large-sized 2SALB problems within relatively very small times. In this study, two heuristics are developed for solving 2SALB Type-I problems.

In the following sections, the heuristics are explained in detail. Finally the performances of the heuristics are compared with other approaches to solve 2SALB, Type-I problems in literature.

4.1 Rolling Horizon Heuristic (RHH) Approach for Type-I 2SALB Problems

The experiments on the mathematical models for Type-I 2SALB problems show that the mathematical model with integer station variables (MM/Int-I) outperforms the model with binary station variables (MM/Bin-I). However, as expected, the qualities of the solutions in a specified time limit decrease as the size of the problem increases. On the other hand, the solutions to the large-sized problems found by MM/Int-I are still reasonable. This gives the idea that the advantages of the mathematical models may be exploited with a heuristic approach. Main logic and the structure of the algorithm that uses the mathematical model are explained in the next section.

4.1.1 Main Logic and Structure of Heuristic Approach

Main logic of this heuristic approach is partitioning the large-sized 2SALB problems into smaller problems decreasing the time and memory requirements. This purpose is achieved by a myopic look to the assembly line and solving the mathematical model to assign tasks only to a specified number of stations at each step. While passing to the next stage, the stations in consideration are updated, like a rolling horizon. The tasks that are not assigned to the specified stations are assigned arbitrarily to a dummy station with no precedence or time constraints.

The heuristic may be summarized as follows:

- Parameters of the heuristics are determined. One of these parameters is the number of stations that the mathematical models will be solved for at each step, in other words, the length of the rolling horizon. The other parameter is the number of stations that the tasks assigned to will be fixed at every step.
- The modified mathematical model for a specified number of stations is solved.
- The solution is investigated. The tasks that are assigned to the specified number of stations are fixed with side, station number and finish time variables. Remaining tasks are still recognized as variables.
- The planning horizon is rolled for the specified number of stations, which is the value of the second parameter. The modified mathematical model is solved with the new variables and fixed values.

The procedure continues until all variables are fixed, in other words, all tasks are assigned. An example progress of the heuristic is displayed in Figure 4.1.

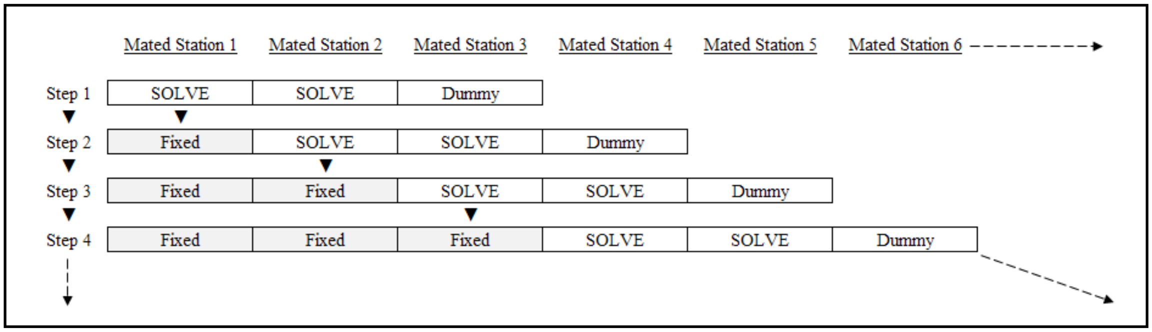

Figure 4.1 Example Progress of RHH

In the example case displayed in Figure 4.1, a rolling horizon length of two is used. In other words, heuristic approach solves a mathematical model that tries to assign tasks to the two stations which are currently in consideration. The mathematical model assigns the remaining tasks to a third station, which serves as a dummy station, regardless of the finish time, side and precedence constraints. The second parameter of the heuristic, number stations that will be fixed at each step, in the displayed example is one. At the end of every step, tasks assigned to the first station of the two stations in consideration are fixed and never changed till the end of the procedure.

Here, a tradeoff appears. Size of the problem decreases as the number of stations filled at each step decrease. Decrease in the size of the problem promises better solutions to the current problem within a specified time limit. On the other hand, as the number stations decrease, the quality of the solutions is threatened since the whole problem cannot be taken into account at once. Hence, determining the length of the planning horizon is another problem that needs to be solved.

In order to better make use of the smaller sized problems that will be solved at each step, using lower bounds on the station numbers play a very important role. Upper bounds cannot be used since the planning horizon moves from the earliest station to the final. In this study, lower bounds are used and updated at each step to make use of the additional information provided at the end of each step in order to create smaller problems. Formulation of the lower bounds is explained in detail in Section 4.1.3.

An important feature of the heuristic is the flexibility on the size of the planning horizon and on the number of stations that will be fixed at each step. This gives the opportunity to find the best solution by the heuristic with respect to the available time and memory limitations. For example, if the model cannot find a feasible solution in the given time limit or returns poor results, the problem size may be reduced by lessening the planning horizon length. On the other hand, if additional time and memory are available, quality of the solution may be increased by increasing the size of the planning horizon.

4.1.2 Additional Notation

The notation used for the mathematical model with integer station variables for Type-I 2SALB problem are also used in this algorithm.

<u>Parameters:</u>

LPH : Length of the planning horizon. In other words, this is the number of stations that the mathematical model will be solved for. An additional dummy station will be added and the remaining tasks will be assigned to the dummy station ignoring the precedence, side and finish time constraints. This value will be fixed at the beginning of the heuristic and at each step a mathematical model for this length will be solved.

APH : Assignment length of the planning horizon. This value needs to be smaller than or equal to LPH. The solution found by the mathematical model for a problem of length LPH does not need to be totally applied. Better solutions may be achieved assigning the tasks to the first APH stations of the solution the problem with LPH stations. Like LPH, APH is fixed at the beginning of the algorithm and the same value is used until all tasks are assigned.

SPH : Starting station number of the planning horizon. This number may be considered as the step number of the algorithm. At each step the mathematical model will be solved for the stations SPH, SPH+1, … SPH+LPH-1. And the number of SPH is increased by APH at the end of each step.

s : A sufficiently small integer

Variables:

a_i : binary variable ($i \in T$)

$$a_i = \begin{cases} 1 & \text{, if task i is assigned to a station in the planning horizon} \\ 0 & \text{, otherwise} \end{cases}$$

4.1.3 Use of Earliest Stations

As explained in the previous section, main idea of this heuristic is to decrease the size of the problem, achieve good and fast solutions for the sub problems and finally, reach the solution to the main problem at the end of a rolling horizon. Use of earliest stations may also well serve this purpose. Extensive use of earliest stations to reduce the size of the sub problems is explained in detail in this section.

Notation:

EST_i : Earliest start time of task i ignoring task indivisibility constraint. The assembly line is assumed to be continuous, regarded as one large station. Hence, EST_i may be greater than the cycle time.

EFT_i : Earliest finish time of task i ignoring task indivisibility constraint. Since the assembly line is assumed to be continuous, EST_i may also be greater than the cycle time.

EST'_i : Earliest start time of task i with task indivisibility constraint. This value is always smaller than or equal to the cycle time.

EFT'_i : Earliest finish time of task i with task indivisibility constraint. This value is always smaller than or equal to the cycle time.

ES_ST_i: Station that the EST_i falls into.

ES_FT_i: Station that the EFT_i falls into.

$ESta_i$: Earliest station that task i can be assigned to.

Formulation:

Earliest stations are calculated with the help of precedence matrix as follows:

$$\mathrm{EST_i} = \max\left\{\max\left\{\mathrm{EST'_p} + \mathrm{t_i} \middle| \mathrm{p} \in \mathrm{P_i^*}\right\} \;,\; \frac{\sum_{i \in P_i} t_i}{2}\right\}$$

$$\mathrm{EFT_i} = \mathrm{EST_i} + \mathrm{t_i}$$

$$\mathrm{ES\_ST_i} = \left\lfloor \frac{\mathrm{EST_i}}{\mathrm{CT}} \right\rfloor + 1$$

$$\mathrm{ES\_FT_i} = \left\lfloor \frac{\mathrm{EFT_i}}{\mathrm{CT}} \right\rfloor + \min\{1, \mathrm{EFT_i} \bmod \mathrm{CT}\}$$

$$\mathrm{EST'_i} = \begin{cases} \mathrm{FT_i} - \mathrm{t_i} & \text{, if task i is already assigned} \\ \begin{cases} \mathrm{EST_i} & \text{, if } \mathrm{ES\_ST_i} = \mathrm{ES\_FT_i} \\ (\mathrm{ES\_FT_i} - 1).\mathrm{CT} & \text{, otherwise} \end{cases} & \text{, if task i is not assigned yet} \end{cases}$$

$$\mathrm{EFT_i^{'}} = \mathrm{EST_i} + \mathrm{t_i}$$

$$\mathrm{ESta_i} = \begin{cases} \mathrm{Sta_i} & \text{, if the task is already assigned} \\ \mathrm{ES\_FT_i} & \text{, if the task is not assigned yet} \end{cases}$$

Earliest stations are calculated recursively using the above calculation. Hence, it requires that a task does not precede a task whose task number is smaller. The formulation may be used without a modification since the requirement is already satisfied by the definition of precedence diagrams.

The calculation of earliest station numbers is dynamic. At the end of each step, the calculation is repeated by fixing the station numbers and finish times of the assigned tasks. This implementation promises better earliest station numbers and further decreases the number of tasks that will be treated as variables for the current planning horizon.

4.1.4 Modified Mathematical Model with Integer Station Variables

The mathematical model is modified to be used in the heuristic. Since the whole problem is not taken into account, the objective of minimizing the length of assembly line cannot be used. Hence, the objective is replaced with minimizing the total idle time of the stations taken into account. Minimizing total idle time is equivalent to maximizing the total of processing times assigned to the stations in consideration.

Objective

$$\max \sum_{i}^{n(T)} t_i \,.\, a_i - s. \sum_{i}^{n(T)} Sta_i \qquad (79)$$

Constraints

$$Sta_i \leq SPH + LPH - a_i \qquad \forall i \in T \quad (80)$$

$$Sta_i \geq (1 - a_i).(SPH + LPH - 1) + 1 \qquad \forall i \in T \quad (81)$$

$$AD_i = Dir_i \qquad \forall i \in L \cup R \quad (82)$$

$$Sta_i + (1 - a_i).(SPH + LPH) \geq Sta_p \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (83)$$

$$(1 - a_i).(SPH + LPH).CT + FT_i \geq (Sta_i - 1).CT + t_i \qquad \forall i \in T \quad (84)$$

$$FT_i \leq Sta_i.CT + (1 - a_i).(SPH + LPH).CT \qquad \forall i \in T \quad (85)$$

$$(1 - a_i).(SPH + LPH).CT + FT_i \geq FT_p + t_i \qquad \forall i \in T \text{ and } \forall p \in P_i \quad (86)$$

$$4.MaxN.CT.(1 - a_i) + 2.MaxN.CT.(1 - Z_{ih}) + MaxN.CT.(AD_h - AD_i) + FT_h \geq FT_i + t_h$$

$$\forall (i,h) \in H \quad (87)$$

$$4.MaxN.CT.(1 - a_i) + 2.MaxN.CT.Z_{ih} + MaxN.CT.(AD_i - AD_h) + FT_i \geq FT_h + t_i$$

$$\forall (i,h) \in H \quad (88)$$

$$Sta_i \in Z \qquad \forall i \in T \quad (89)$$

$$AD_i \in \{0,1\} \qquad \forall i \in T \quad (90)$$

$$Z_{ih} \in \{0,1\} \qquad \forall (i,h) \in H \quad (91)$$

$$a_i \in \{0,1\} \qquad \forall i \in T \quad (92)$$

The objective (79) maximizes the sum of processing times of tasks assigned to the stations in consideration, i.e. minimizes the idle time of the tasks that are assigned to these stations. Second part of the objective is needed at the final steps of the algorithm if the length of the planning horizon, LPH, is greater than one station. Since the sum of station numbers is multiplied with a sufficiently small integer, it serves as a secondary objective and tries to compress the remaining tasks to the earliest stations if possible at the final steps of the algorithm. Constraint (80) and (81) are used for assigning the tasks with a_i=1 into the planning horizon and assigning the remaining tasks to the dummy station. Constraints (82) satisfies the side constraints of left and right tasks. Constraint (83) states that tasks cannot be assigned to an earlier station than the stations its predecessors are assigned to and this constraint only applies to the tasks with a_i=1. Constraints (84) and (85) ensure that FT_i lies between the starting time of the station that the task is assigned and the finishing time of that station. Constraint (86) states that tasks with precedence relations cannot be operated simultaneously. Constraints (87) and (88) prevent tasks with no precedence relation and which are assigned to the same side from overlapping if they are assigned in the current step. Constraints (89), (90), (91) and (92) state the types of the variables.

4.1.5 Algorithm

The steps of the algorithm are explained below:

Step 0. Set a_i = 0 for all tasks and set SPH = 1. Determine the heuristic parameters LPH and APH.

Step 1. Calculate earliest station numbers.

If the earliest station number of task i is larger than or equal to (SPH + LPH), eliminate the task from the problem by carrying out necessary fixations:

- fix a_i = 0 (discard the task from the objective)
- fix Sta_i = SPH + LPH (assign the task to the dummy station)

Do not fix variables of the remaining tasks. Also remove fixations of earlier stages from these tasks.

Step 2. Solve the modified mathematical model for the fixed LPH, fixed APH and current SPH.

The fixed values of variables in Step 1 are not violated.

Step 3. Get the solution from the mathematical model and investigate Sta_i values.

If $Sta_i \le SPH + APH - 1$,

- fix $a_i = 1$
- fix Sta_i, AD_i and FT_i to the values found in the current solution.

Unlike Step 1, these values will remain fixed until the end of the algorithm.

Step 4. Calculate the sum of the values a_i for all tasks.

- If the sum is equal to the number of tasks, i.e. if all tasks are assigned, STOP. The solution to the final mathematical model is the solution of the algorithm to the given problem.
- Otherwise, set SPH = SPH + APH. Go to Step 1.

The results of the RHH are given later in this chapter after the second heuristic is introduced.

4.2 Extended Multiple Rule Heuristic (EMRH) Approach for Type-I 2SALB Problems

Inspired by the multiple rule heuristic approach (Boctor, 1995), another heuristic approach to generate fast and good solutions to 2SALB Type-I problems is developed. The rules used by the multiple rule heuristic approach are given in Section 2.2. Heuristic approach proposed in this study may be regarded as an extended version of the multiple rule heuristic approach. Determination of the rules for task selection is based on numerous experiments. The algorithm is programmed in Microsoft Visual C# 2008 Express Edition and tested on Core2 Quad, 2.84 GHz, 2.00 GB RAM personal computer.

4.2.1 Main Logic and Structure of Heuristic Approach

The heuristic tries to assign one task at each step of the algorithm. This kind of assignment definitely generates fast solutions. The quality of the solutions depends on the rules followed for task selection and task assignment procedures. Since this is the most critical part of the heuristic, many experiments on different problems are carried out. The rules are determined by these observations. However, it is very hard to claim the best rules. The best combination of rules for a particular problem may result in a poor result for another problem. Since the algorithm works fast enough, it is decided to try various combinations of selection and assignment rules on a single run. The results for all combinations followed are summarized at the end of the algorithm. Hence, solutions of this heuristic approach can also be regarded as an experiment for observing which combination of selection and assignment criteria performs better.

The weakness of this kind of heuristic approaches is that they are stacked to the rules determined at the beginning. As described above, this heuristic tries to overcome this weakness by using different combinations of selection and assignment rules. However, the weakness still exists since a combination of rules may be better at one stage of the algorithm while another combination may be better on another. This approach also tries to overcome this weakness by generating random rules at the beginning of each step. This property of EMRH creates diversity and tries to handle the cases that could not be handled by the rules determined by the observations to the experiments.

The algorithm may be summarized as follows:

- Generate the set of available tasks (the tasks whose immediate predecessors are already assigned) at the beginning of each step
- Apply four selection rules to select the best task from the available task set
- Apply three assignment rules to assign the selected task

The rules are described in detail in the following sections.

4.2.2 Selection Rules

Eight selection rules are developed in this approach. Each algorithm uses four of them to select the task that will be assigned at each step. The rules are applied in a hierarchical way. Hence, the order of the rules is also important.

Developed rules are described below:

<u>Minimum Station (MinSta)</u>: This is the minimum station number that the task can be assigned to. This rule is surely the most prior selection rule in order to minimize the total number of stations. Hence, in all combinations of rules, this is used as the first selection rule.

<u>Minimum Start Time (MinST)</u>: This is the start time of the candidate task in the station. In calculation of this value, the finish time of the predecessors of the task is taken into account.

Maximum Finish Time (MaxFT): This is the finish time of the candidate task in the station if selected and assigned. This rule tries to assign the task with the longest processing time while possible.

Side Constrained (Constrained): This rule selects left and right tasks among the candidate tasks and eliminates tasks without side constraints. This rule also tries to relax the later stages of the problem.

Minimum Idle (MinIdle): This is rule is the most complex one in calculations. Idle times are calculated in two parts. First part is the difference between the start time of the selected task and finish time of the previous task assigned to the same station. The second part calculates the idle time after the completion of the selected task. For this calculation, the algorithm simulates the further step. If no tasks can be assigned to the station that the selected task is to be assigned, the difference between the cycle time and the finish time of the selected task is also regarded as idle time. For example, if the selected task finishes at time 4 in the station with cycle time 5 and if there are no tasks that can be assigned to the 1 unit of time after this task, this difference is also regarded as idle.

Maximum Immediate Successors (MaxImmSuc): This rule weights the tasks with respect to their precedence relations. However, this rule looks one stage ahead, since it only counts the immediate successors.

Maximum Successors (MaxSuc): This is the number of all successors of the candidate tasks. The rule selects the task with the maximum number of successors, i.e. with the maximum $n(S^*_i)$.

Maximum Task Time (MaxTT): With this rule it is aimed to assign the longest tasks first. This rule helps to create a balance in which longest tasks take the first places in the stations to reduce the risk of leaving idle times.

36 of the possible combinations of these rules each containing four rules are selected for the algorithm with respect to the experiments. The first rule in all of these combinations is MinSta. Each combination is called a sub-algorithm (SA). All sub-algorithms used by EMRH in a single run are summarized in Table 4.1.

Table 4.1 Combinations of Selection Rules used by EMRH

	Rule 1	Rule 2	Rule 3	Rule 4
SA-1	MinSta	MinST	MinIdle	MaxSuc
SA-2	MinSta	MinIdle	MinST	MaxSuc
SA-3	MinSta	MinST	MaxSuc	MinIdle
SA-4	MinSta	MinIdle	MaxSuc	MinST
SA-5	MinSta	MaxSuc	MinIdle	MinST
SA-6	MinSta	MaxSuc	MinST	MinIdle
SA-7	MinSta	MinST	MinIdle	MaxFT
SA-8	MinSta	MinIdle	MaxFT	MinST
SA-9	MinSta	MinIdle	MaxFT	MaxSuc
SA-10	MinSta	MinIdle	MaxSuc	MaxFT
SA-11	MinSta	MinST	MinIdle	MaxImmSuc
SA-12	MinSta	MinIdle	MinST	MaxImmSuc
SA-13	MinSta	MinST	MaxImmSuc	MinIdle
SA-14	MinSta	MinIdle	MaxImmSuc	MinST
SA-15	MinSta	MaxImmSuc	MinIdle	MinST
SA-16	MinSta	MaxImmSuc	MinST	MinIdle
SA-17	MinSta	MinIdle	MaxFT	MaxImmSuc
SA-18	MinSta	MinIdle	MaxImmSuc	MaxFT
SA-19	MinSta	MinIdle	MaxTT	MaxSuc
SA-20	MinSta	MinIdle	MaxSuc	MaxTT
SA-21	MinSta	MaxSuc	MinIdle	MaxTT
SA-22	MinSta	MaxSuc	MaxTT	MinIdle
SA-23	MinSta	MaxTT	MinIdle	MaxSuc
SA-24	MinSta	MinIdle	MaxTT	MaxImmSuc
SA-25	MinSta	MaxTT	MaxImmSuc	MinIdle
SA-26	MinSta	MinIdle	MaxImmSuc	MaxTT
SA-27	MinSta	MaxImmSuc	MinIdle	MaxTT
SA-28	MinSta	MaxImmSuc	MaxTT	MinIdle
SA-29	MinSta	MinIdle	Constrained	MaxSuc
SA-30	MinSta	MinIdle	Constrained	MaxImmSuc
SA-31	MinSta	MinST	Constrained	MaxSuc
SA-32	MinSta	MinST	Constrained	MaxImmSuc
SA-33	MinSta	MinIdle	MaxSuc	Constrained
SA-34	MinSta	MinIdle	MaxImmSuc	Constrained
SA-35	MinSta	MinST	Constrained	MaxSuc
SA-36	MinSta	MinST	MaxImmSuc	Constrained

4.2.3 Assignment Rules

A variable called 'Preferred Side' is stored through the algorithm for each E task. These values are updated at each step depending on the selection rules. The side satisfying the selection rule is determined as the preferred side of the task for the current step. For example, if the candidate task may be assigned to station 2 if assigned to left and may be assigned to station 3 if assigned to right, then the preferred side with respect to MinSta rule is updated as left. Preferred sides of E tasks are returned to either at the beginning of each step.

Assignment rules are needed when the selected task is an E-task and the preferred side of the task is also E. In other words, these rules are used whenever assigning the task to the right and left stations are indifferent according to selection rules. Three assignment rules are developed to handle such cases:

<u>Remaining All (a)</u>: Processing times of all unassigned left tasks and task times of all unassigned right tasks are summed. Selected task is assigned to the side with minimum sum of remaining task time. This rule tries to leave space to the side with maximum total of remaining processing times.

<u>Remaining 2 Steps (b)</u>: Processing times of the tasks in the available task set and their immediate successors are summed for left and right tasks. Selected task is assigned to the side with minimum sum of remaining task time.

<u>Remaining 1 Step (c)</u>: Processing times of the tasks in the available task set are summed for left and right tasks. Selected task is assigned to the side with minimum sum of remaining task time.

36 combination of selection rules are used with each of these three assignment rules. Hence, the algorithm actually solves 108 defined algorithms in a single run. From this point on the sub-algorithms are named with the extension of the assignment rule sign. For instance, SA-1-a uses the selection rules listed in Table 4.10 and the assignment rule Remaining All.

4.2.4 Random Rules

In addition to the 108 defined combinations of selection and assignment rules, the algorithm further tries to provide more diversity using a random algorithm.

Random algorithm generates a combination of four selection rules among the eight developed rules. Then, an assignment rule among the three proposed assignment rules is selected. A task is selected from the available task set with respect to the random selection rules and assigned according to the random assignment rule. At the beginning of the next step, this process is repeated. Hence, the procedure is not stacked to one combination of rules through the algorithm, but uses different rules at each assignment.

Number of random trials is entered before the algorithm is run. The current best solution is kept throughout the algorithm. Only the results that are as good as the current best solution are stored in order to reduce memory requirement. The best solutions found by the random algorithms are summarized at the end of the algorithm together with the solutions of the 108 sub-algorithms.

4.2.5 Algorithm

Additional notation used in the algorithm is described below:

<u>Additional Notation:</u>

NL : Latest opened left station number.

NR : Latest opened right station number.

FT_L(j) : Finish time of the latest task in the left station of the j^{th} mated-station.

FT_R(j) : Finish time of the latest task in the right station of the j^{th} mated-station.

a(i) : Binary variable indicating whether task i is assigned (i ∈ T)

$$a_i = \begin{cases} 1 & \text{, if task i is assigned} \\ 0 & \text{, otherwise} \end{cases}$$

FT(i) : Finish time of task i (i ∈ T)

S(i) : The side that task i is assigned to, L or R (i ∈ T)

PS(i) : Preferred side of unassigned task i according to the current selection or assignment rule in consideration (i ∈ E)

$MS_D(i)$: Minimum station number that task i may be assigned to that is on the side S, where D ∈ {L,R} (i ∈ T). If the considered task can be assigned to the finally opened station, the value of this variable is set to NL or NR according to the side constraint of the task. Otherwise this variable is set to NL+1 or NR+1. For E tasks, minimum station numbers for both L and R station are calculated.

$MST_D(i)$: Earliest start time of the considered task for the earliest station on the side S, where D ∈ {L,R}, that it can be assigned to (i ∈ T). For E tasks minimum start times for both L and R stations are calculated. In the calculation of this variable, FT_L(i) and FT_R(i) are taken into account together with the cycle time and the latest finish time of the immediate predecessor tasks of the considered task.

IT_D (i) : This is the sum of the idle time left just before the starting time of the task and the idle time after the finish time of the task to the end of the cycle time (i ∈ T) for the station on the side D, where D ∈ {L,R}. First part of the summation time may occur due to precedence relations. Second part requires more complex calculations. It is assumed that the task in consideration is assigned and all the variables for the next step are calculated. If there are no tasks that may be assigned to the same station as the task in consideration, second part of the summation is calculated as the gap between the cycle time and assumed finish time of the task. Otherwise, this part of the summation is set as zero. This procedure tries to correct the miscalculations of resulting from ignoring the idle times that may occur after the considered task is finished. For E tasks, this variable is calculated for both L and R stations.

The steps of the algorithm are summarized below:

Step 0. Open the first mated-station by setting:

NL = 1

NR = 1

FT_L(1) = 0

FT_R(1) = 0

Step 1. Generate the set of available tasks, Set_0. An available task is an unassigned task whose immediate predecessors are all assigned.

$$Set\_0 = \{i \in T | a_i = 0 \text{ and } a_p = 1 \; \forall i \in P_i^*\}$$

- If $n(Set\_0) = 0$, STOP
- Else go to Step 2.

Step 2. Calculate the variables MS(i), MST(i) and IT(i) for each task in Set _0. Set PS(i) = E for all E tasks in the set. Set PS(i) of the L and R tasks to their side constraints.

For E tasks,

- If $MS_L(i) > MS_R(i)$, set PS(i) = R
- If $MS_L(i) < MS_R(i)$, set PS(i) = L
- Otherwise, preferred side remains as E.

Step 3. If n(Set _0) = 1, assign the task:

- If PS(i) = L, update the variables as below:

  $$FT\_L = \begin{cases} MST_L(i) + t_i & \text{, if } NL = MS_L(i) \\ t_i & \text{, otherwise} \end{cases}$$

  $NL = MS_L(i)$

  a(i) = 1

  Return to Step 1.
- If PS(i) = R, update the variables as below:

  $$FT\_R = \begin{cases} MST_R(i) + t_i & \text{, if } NR = MS_R(i) \\ t_i & \text{, otherwise} \end{cases}$$

  $NR = MS_R(i)$

  a(i) = 1

  Return to Step 1.
- If PS(i) = E, update PS(i) according to the currently used assignment rule. If a tie occurs, update PS(i) to L or R randomly. Then, repeat this step.

Otherwise, go to Step 4.

Step 4. Generate Set _1 among the tasks in Set _0 and update PS(i) of E tasks as the side that satisfies the first selection rule. It may remain E, if the rule is satisfied by both sides.

$$\text{Set }\_1 = \{\text{i} \in \text{Set }\_0 | \text{i satisfies the first selection rule}\}$$

If n(Set _1) = 1, assign the task:

- If PS(i) = L, update the variables as below:

  $$\text{FT\_L} = \begin{cases} \text{MST}_\text{L}(\text{i}) + \text{t}_\text{i} & , \text{if NL} = \text{MS}_\text{L}(\text{i}) \\ \text{t}_\text{i} & , \text{otherwise} \end{cases}$$

  NL = MS_L(i)

  a(i) = 1

  Return to Step 1.
- If PS(i) = R, update the variables as below:

  $$\text{FT\_R} = \begin{cases} \text{MST}_\text{R}(\text{i}) + \text{t}_\text{i} & , \text{if NR} = \text{MS}_\text{R}(\text{i}) \\ \text{t}_\text{i} & , \text{otherwise} \end{cases}$$

  NR = MS_R(i)

  a(i) = 1

  Return to Step 1.
- If PS(i) = E, update PS(i) according to the currently used assignment rule. If a tie occurs, update PS(i) to L or R randomly. Then, repeat this step.

Otherwise, go to Step 5.

Step 5. Generate Set _2 among the tasks in Set _1 and update PS(i) of E tasks as the side that satisfies the first selection rule. It may remain E, if the rule is satisfied by both sides.

$$\text{Set }\_2 = \{\text{i} \in \text{Set }\_1 | \text{i satisfies the second selection rule}\}$$

If n(Set _2) = 1, assign the task:

- If PS(i) = L, update the variables as below:

$$\text{FT\_L} = \begin{cases} \text{MST}_\text{L}(\text{i}) + \text{t}_\text{i} & \text{, if NL} = \text{MS}_\text{L}(\text{i}) \\ \text{t}_\text{i} & \text{, otherwise} \end{cases}$$

NL = MS_L(i)

a(i) = 1

Return to Step 1.

- If PS(i) = R, update the variables as below:

$$\text{FT\_R} = \begin{cases} \text{MST}_\text{R}(\text{i}) + \text{t}_\text{i} & \text{, if NR} = \text{MS}_\text{R}(\text{i}) \\ \text{t}_\text{i} & \text{, otherwise} \end{cases}$$

NR = MS_R(i)

a(i) = 1

Return to Step 1.

- If PS(i) = E, update PS(i) according to the currently used assignment rule. If a tie occurs, update PS(i) to L or R randomly. Then, repeat this step.

Otherwise, go to Step 6.

Step 6. Generate Set _3 among the tasks in Set _2 and update PS(i) of E tasks as the side that satisfies the first selection rule. It may remain E, if the rule is satisfied by both sides.

$$\text{Set \_3} = \{\text{i} \in \text{Set \_2} | \text{i satisfies the third selection rule}\}$$

If n(Set _3) = 1, assign the task:

- If PS(i) = L, update the variables as below:

$$\text{FT\_L} = \begin{cases} \text{MST}_\text{L}(\text{i}) + \text{t}_\text{i} & \text{, if NL} = \text{MS}_\text{L}(\text{i}) \\ \text{t}_\text{i} & \text{, otherwise} \end{cases}$$

NL = MS_L(i)

a(i) = 1

Return to Step 1.

- If PS(i) = R, update the variables as below:

$$\text{FT\_R} = \begin{cases} \text{MST}_\text{R}(\text{i}) + \text{t}_\text{i} & \text{, if NR} = \text{MS}_\text{R}(\text{i}) \\ \text{t}_\text{i} & \text{, otherwise} \end{cases}$$

NR = MS_R(i)

a(i) = 1

Return to Step 1.

- If PS(i) = E, update PS(i) according to the currently used assignment rule. If a tie occurs, update PS(i) to L or R randomly. Then, repeat this step.

Otherwise, go to Step 7.

Step 7. Generate Set _4 among the tasks in Set _3 and update PS(i) of E tasks as the side that satisfies the first selection rule. It may remain E, if the rule is satisfied by both sides.

$$\text{Set\_4} = \{\text{i} \in \text{Set\_3} | \text{i satisfies the fourth selection rule}\}$$

Assign the first task in Set _4:

- If PS(i) = L, update the variables as below:

  $$\text{FT\_L} = \begin{cases} MST_L(i) + t_i & \text{, if NL} = MS_L(i) \\ t_i & \text{, otherwise} \end{cases}$$

  NL = MS_L(i)

  a(i) = 1

  Return to Step 1.
- If PS(i) = R, update the variables as below:

  $$\text{FT\_R} = \begin{cases} MST_R(i) + t_i & \text{, if NR} = MS_R(i) \\ t_i & \text{, otherwise} \end{cases}$$

  NR = MS_R(i)

  a(i) = 1

  Return to Step 1.
- If PS(i) = E, update PS(i) according to the currently used assignment rule. If a tie occurs, update PS(i) to L or R randomly. Then, repeat this step.

This seven step algorithm is in fact one of the 108 sub-algorithms that are solved by EMRH for a single run. Whenever EMRH reaches a solution for a sub-algorithm, the details of the solution are stored in the hard disk to reduce the virtual memory requirements. A report displaying the number of left and right stations found by each of the 108 sub-algorithms is created at the end of the EMRH run. Moreover, random algorithms may be solved for a specified number of times after the 108 sub-algorithms are solved. Random algorithms follow the same procedure as the algorithm described above except that generates four selection rules and one assignment rule whenever Step 2 is visited. EMRH keeps the best current solution throughout the random algorithms and only saves the detailed solutions which are at least as good as the current best solution.

The algorithm uses minimum station rule as the first rule in all of the sub-algorithms. Use of this rule as the first selection criterion guarantees that the objective of the heuristic is to minimize the length of the line, in other words, to minimize the number of mated-stations. This rule with the definition of the variables NL and NR also serves as a secondary objective of minimizing the number of stations.

Figure 4.2 displays the flow chart of a sub-algorithm of EMRH. With this figure, hierarchical use of the selection rules may be better understood.

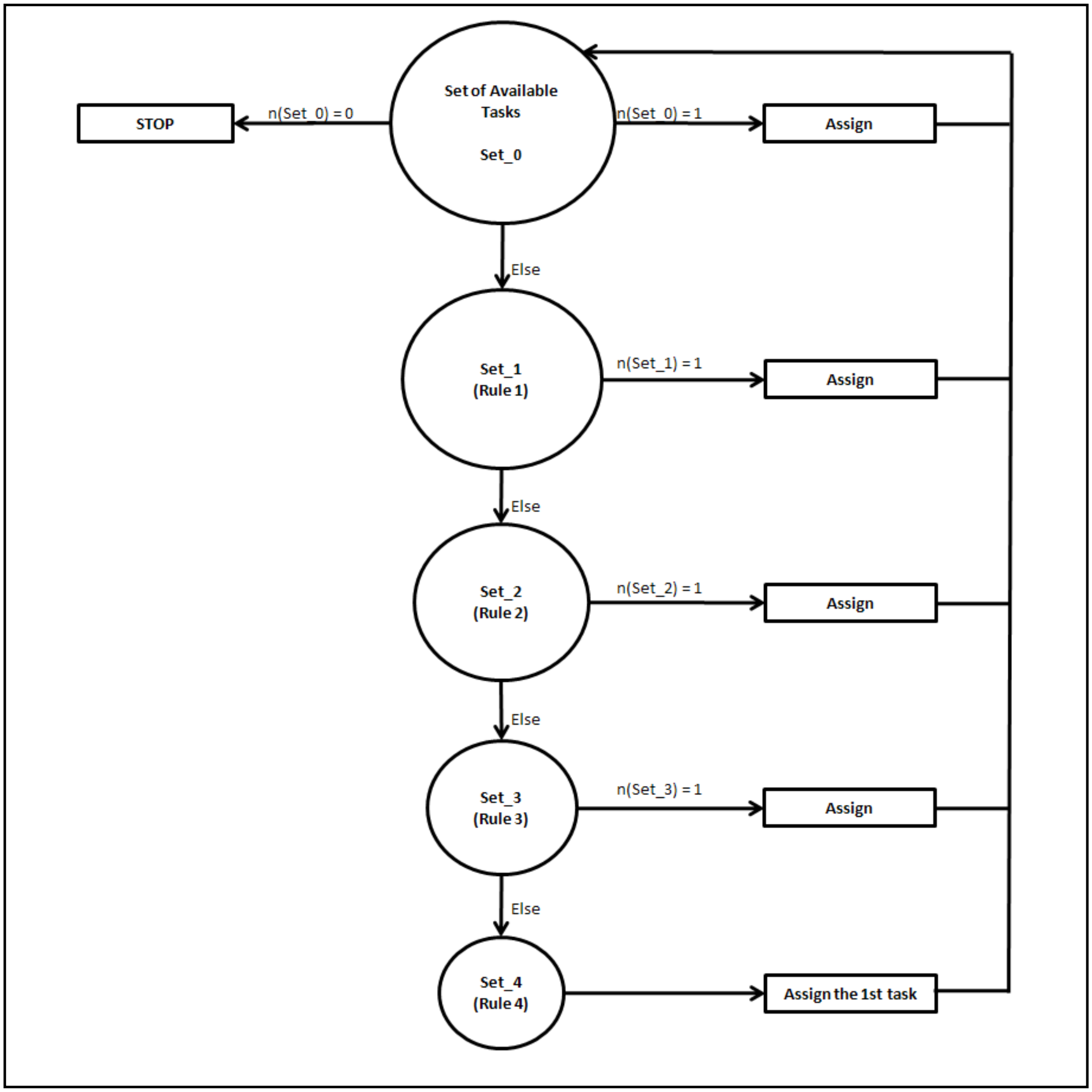

Figure 4.2 Flowchart of EMRH

4.3 Performance Comparison of RHH, EMRH and Other Heuristics in Literature

Three large-sized problems focused in the literature are used for testing the heuristics: 148-task problem (Bartholdi, 1993), 65-task problem and 205-task problem (Lee et al., 2001). These problems will be named as P148, P65 and P205 respectively. Data of the problems may be found in Appendix A. Solutions to these large-sized problems with the same objective are proposed by the ant-colony optimization algorithm (ACO) by Baykasoglu and Dereli (2008), branch-and-bound (B&B) algorithm proposed by Wu et al. (2008) and another ant-colony-optimization algorithm (2-ANTBAL) proposed by Simaria and Vilarinho (2009). Solutions proposed by these authors are used to test the performance of the heuristic approaches introduced in this study. Simaria and Vilarinho (2009) proposed the minimum, maximum and average number of stations of 10 runs of 2-ANTBAL. In this study, minimum values are used for comparison.

Three measures are used to test the performance of the solutions:

N : Number of mated-stations, i.e. length of the assembly line.

n : Number of stations, $n \in [2.N-1, 2.N]$.

This measure is not used to compare the performance of RHH, since it is not included in the objective of the heuristic approach.

Solution Time : Required time to reach the solution. This measure is used to compare RHH and EMRH only.

Lower bounds on the number of mated-stations and lower bounds on the number of stations are calculated using the formulations proposed by Wu et al. (2008):

LB_N and LB_n are calculated by the formulations proposed by Wu et al. (2008):

$$S_L = \left\lceil \left(\sum_{i \in L} t_i \right) / CT \right\rceil$$

$$S_R = \left\lceil \left(\sum_{i \in R} t_i \right) / CT \right\rceil$$

$$S_E = \left\lceil \frac{\max\left(\left(\sum_{i \in E} t_i - \left((S_L + S_R).CT - \left(\sum_{i \in L} t_i + \sum_{i \in R} t_i\right)\right)\right), 0\right)}{CT} \right\rceil$$

$$LB_N = \max(S_L + S_R) + \left\lceil \frac{\max\left((S_E - |S_L - S_R|), 0\right)}{2} \right\rceil$$

$$LB_n = S_L + S_R + S_E$$

Solutions for 65-task problem are summarized in Table 4.2.

Table 4.2 of RHH, EMRH, ACO, B&B and 2-ANTBAL for P65

Cycle Time	381	490	544
LB for N	7	6	5
Optimal N	7	6	5
LB for n	14	11	10
Optimal n	14	11	10
ACO			
N	8	6*	5*
n	15	12	10*
B&B			
N	7*	6*	5*
n	14*	11*	10*
2-ANTBAL			
N	7*	6*	5*
n	14*	12	10*
RHH			
N	7*	6*	5*
CPU time(s)	1542.87	1209.86	1200.10
EMRH			
N	7*	6*	5*
n	14*	11*	10*
CPU time(s)	3.50	3.50	3.50

Asterisk sign, *, indicates that the solution found is optimal. Number of stations, n, is not given for RHH since it aims only to minimize the length of the assembly line, N.

From Table 4.2, it can be seen that B&B, 2-ANTBAL, RHH and EMRH perform best reaching optimal solutions with respect to the objective of minimizing N for all of the problems. With respect to the objective of minimizing n, B&B and EMRH perform best again finding the optimal solutions for all of the problems.

Solutions for 148-task problem are summarized in Table 4.3.

Table 4.3 Performances of RHH, EMRH, ACO, B&B and 2-ANTBAL for P148

Cycle Time	204	255	306	357	408	459	510
LB for N	13	11	9	8	7	6	6
Optimal N	13	11	9	8	7	6	6
LB for n	26	21	17	15	13	12	11
Optimal n	26	21	n/a	15	13	12	11
ACO							
N	13*	11*	9*	8*	7*	6*	6*
n	26*	21*	18	15*	14	12*	11*
B&B							
N	13*	11*	n/a	8*	7*	6*	6*
n	26*	21*	n/a	15*	13*	12*	11*
2-ANTBAL							
N	13*	11*	9*	8*	7*	6*	6*
n	26*	21*	18	15*	14	12*	11*
RHH							
N	13*	11*	9*	8*	7*	6*	6*
CPU time(s)	10685.36	5920.06	4798.11	4193.09	3601.01	3599.16	2995.88
EMRH							
N	13*	11*	9*	8*	7*	6*	6*
n	26*	21*	17*	15*	13*	12*	11*
CPU time(s)	17.00	17.00	17.00	18.00	18.00	18.00	18.00

Asterisk sign, *, indicates that the solution found is optimal. The optimal solution with respect to the objective of minimizing the number of stations to the 148-task with cycle time 306 is not known yet and the solution to this specific problem by B&B is not proposed by Wu et al. (2008). Hence corresponding entries are “n/a” in Table 4.3.

For the problem with 148 tasks, five of the heuristic approaches managed to find optimal line lengths for all of the problems. With respect to the secondary objective, B&B and EMRH performed best finding optimal number of opened stations to all of the problems. EMRH also managed to introduce the optimal solution for the problem with cycle time 306 for the first time.

Solutions for 205-task problem are summarized in Table 4.4.

Table 4.4 Performances of RHH, EMRH, ACO and 2-ANTBAL for P205

Cycle Time	1133	1322	1510	1699	1888
LB for N	11	9	8	7	7
Optimal N	11	n/a	n/a	n/a	7
LB for n	21	18	16	14	13
Optimal n	n/a	n/a	n/a	n/a	13
ACO					
N	12	11	9	9	8
n	24	22	18	18	15
2-ANTBAL					
N	11*	10	9	8	7*
n	22	20	17	15	13*
RHH					
N	11*	10	8*	7*	7*
CPU time(s)	5991.31	5397.54	4201.21	3603.67	3599.77
EMRH					
N	11*	9*	8*	7*	7*
n	22	18*	16*	14*	13*
CPU time(s)	27	27	29	25	28

Table 4.4 (continued)

Cycle Time	2077	2266	2454	2643	2832
LB for N	6	6	5	5	5
Optimal N	6	6	5	5	5
LB for n	12	11	10	9	9
Optimal n	12	n/a	10	n/a	n/a
ACO					
N	7	6*	6	6	5*
n	14	12	12	11	10
2-ANTBAL					
N	6*	6*	5*	5*	5*
n	12*	12	10*	10	10
RHH					
N	6*	6*	5*	5*	5*
CPU time(s)	6001.35	4499.64	2999.70	3600.26	2399.93
EMRH					
N	6*	6*	5*	5*	5*
n	12*	12	10*	10	10
CPU time(s)	33	34	31	28	27

Asterisk sign, *, indicates that the solution found is optimal. B&B solutions to P205 are not proposed by Wu et al. (2008), and therefore, this heuristic is discarded from the comparison for P205.

Problems with 10 different cycle times are solved with the heuristic approaches.

With respect to the primary objective of minimizing the line length, ACO achieved to find optimal solutions for two of the problems while 2-ANTBAL achieved to find optimal solutions for seven of the problems. On the other hand, RHH achieved nine optimal results while EMRH managed to find optimal solutions to all of the problems. Three of these optimal solutions are introduced for the first time.

With respect to the secondary objective, ACO could not manage to find optimal solutions while 2-ANTBAL achieved to find optimal solutions for three of the problems. EMRH manage to verify optimality for six of the problems and found at least as good solutions as the other approaches did for the remaining four problems. Three of the optimal solutions are introduced for the first time.

Complete solutions found by EMRH to three of the large-sized problems are given in Appendix D.

With respect to the solution times, EMRH outperforms RHH. This is the expected result according to the structures of the heuristic approaches. EMRH managed to find solutions in at most 34 seconds for the large-sized literature problems. However, when compared with the mathematical models to solve Type-I 2SALB problems, RHH managed to find very good solutions in reasonable solution times. Well-known large-sized problems studied in the literature, EMRH managed to find the best results. However, the rules may still fail to perform well in unpredicted cases. In such cases, RHH may be used to exploit the advantages of mathematical models and find good solutions in reasonable solution times.

4.4 Performance Tests of EMRH with respect to Side Freedom

In order to observe the performances of EMRH for large-sized problems with different numbers of side constraints, experimental problems are generated from the 205-task problem (P205) proposed by Lee et al. (2001) and solved by the proposed heuristic. In order to define this characteristic of 2SALB problems, side freedom (SF), ratio of the number of E tasks to the number of all tasks, is used:

$$\mathrm{SF} = \frac{\mathrm{n(E)}}{\mathrm{n(T)}} \text{ x } 100\%$$

Precedence relations and processing times of P205 are used for the experimental problems. Initially, all tasks are set to be E resulting in the problem with SF = 100%. This problem is P205 with all side constraints relaxed. While generating the other problems, 10% of the E tasks in the previous problem are randomly selected and updated as L or R tasks arbitrarily. Hence, each problem is a partially relaxed version of the problems that generated after the problem. With SF ranging from 100% to 0%, 11 problems are generated. Each problem is solved for three different cycle times, 1133, 1888 and 2643.

Lower bounds on the number of mated-stations (LB_N) and lower bounds on the number of stations (LB_n) are again calculated with the formulation proposed by Wu et al. (2008).

For comparing the solutions, line length gap and station number gap are used:

$$\text{Line Length Gap} = \frac{(\text{Solution} - \text{LB}_\text{N})}{\text{LB}_\text{N}} \text{ x } 100\%$$

$$\text{Station Number Gap} = \frac{(\text{Solution} - \text{LB}_\text{n})}{\text{LB}_\text{n}} \text{ x } 100\%$$

33 problems are solved by EMRH and the best solutions found by the 108 sub-algorithms are used for comparing with the lower bounds. For all of the problems, solutions equal to LB_N are reached. Averages of deviations from the lower bounds on the number of stations are displayed in Figure 4.3

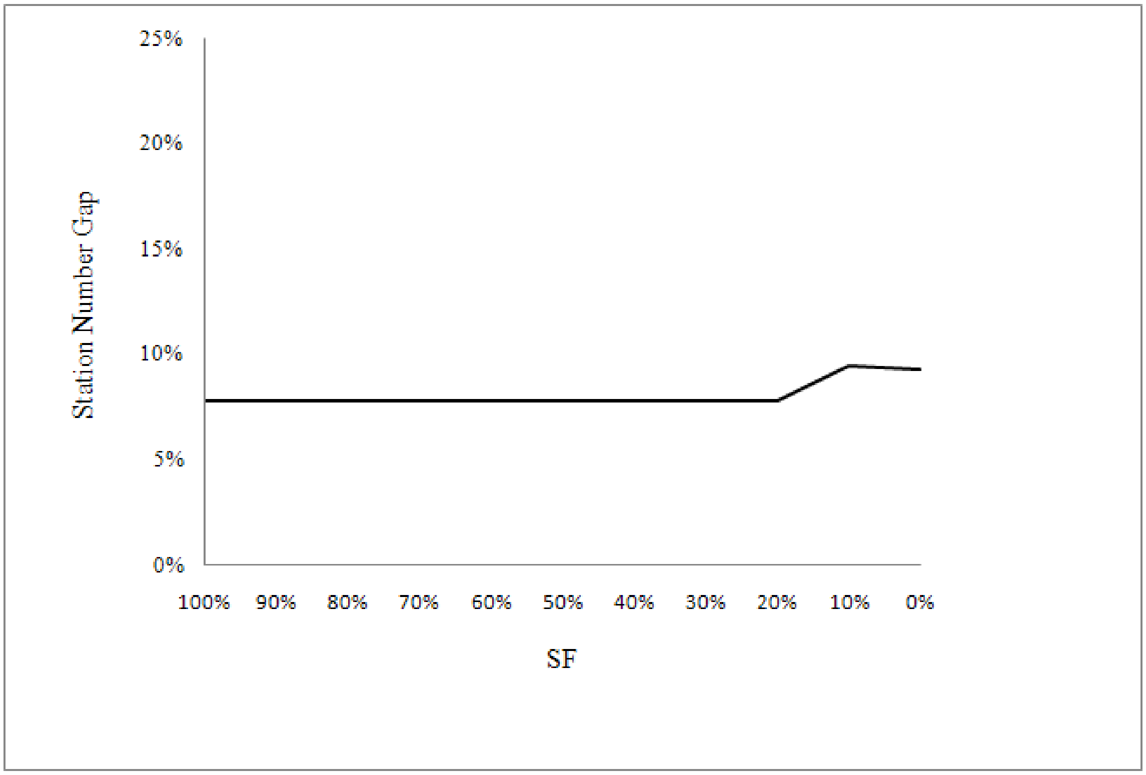

Figure 4.3 Station Number Gap Performance of EMRH

From Figure 4.3, EMRH keeps finding solutions as good as the solution found for the most relaxed problem until SF is equal to 10%. For the remaining two problems, station number gap slightly increases.

EMRH tries to overcome the additional difficulty resulting from additional side constraints with different sub-algorithms. Hence, it is difficult to comment on the effect of side freedom from Figure 4.3. In order to observe the effect of this parameter, performances of sub-algorithms need to be investigated. Figure 4.4 displays the number of sub-algorithms that achieves the best solutions found by EMRH for each problem.

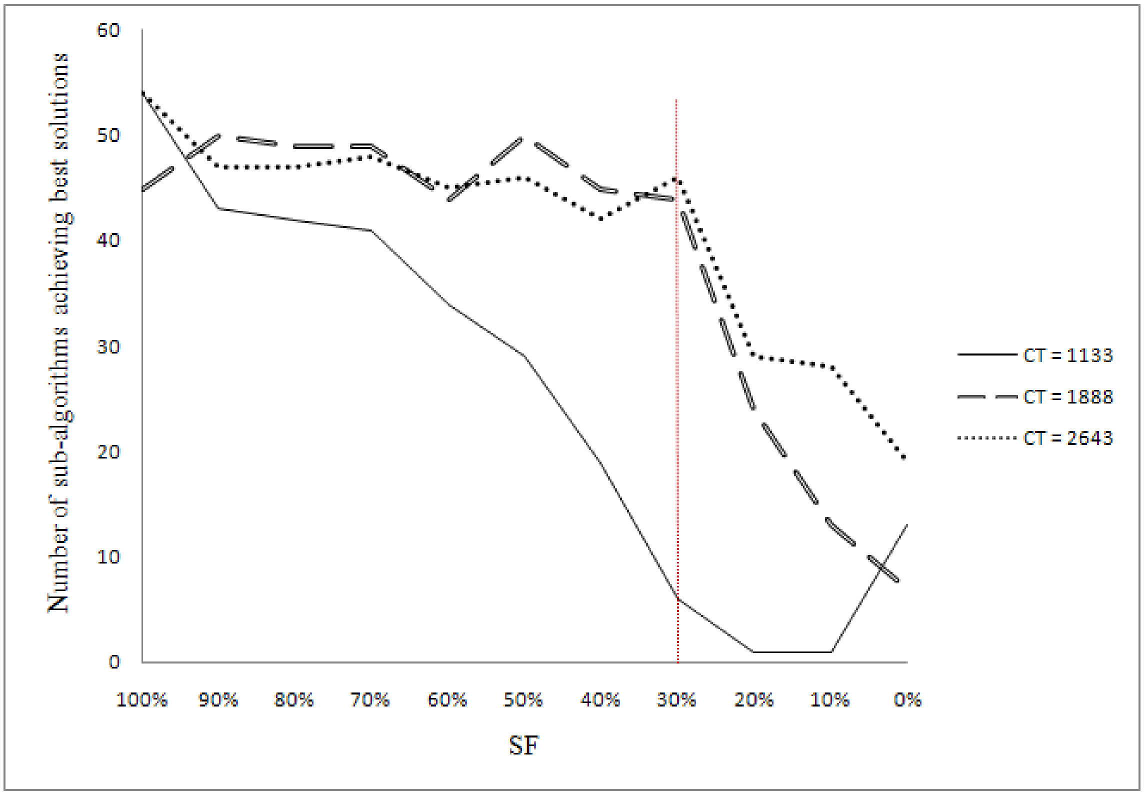

Figure 4.4 Performances of Sub-Algorithms of EMRH

Since EMRH found the same solutions to almost all of the problems with different side freedoms, number of sub-algorithms that achieve the best solution of EMRH indicates the performance of sub-algorithms. Figure 4.4 shows that a smaller number of sub-algorithms manage to find the best solution as SF decreases. For the problems with CT = 1133, this number even decreases to one. The increase in the last problem is due to the increase in the lower bound of the problem. In other words, with the addition of new side constraints, a new station is opened and the problem is relaxed resulting in a more number of sub-algorithms achieving the best solution. Hence, this increase should not be considered in the analysis. For problems with cycle times 1888 and 2643, number of sub-algorithms achieving the best solution is almost stable until

SF equals 30%. For remaining problems, number of sub-algorithms achieving the best solution greatly decreases.

CHAPTER 5

CONCLUSION

This study focuses on solving two-sided assembly line balancing (2SALB) problems. This problem is very recent to the literature and very limited number of studies exists concerning 2SALB. Existing studies generally propose a mathematical model and a heuristic approach. Mathematical models are verified with small-sized literature problems; however, solutions for large-sized problems have not been introduced yet. Similarly, this study also proposes mathematical models to solve 2SALB problems. Two mathematical models are developed for each of type-I and type-II problems, one with binary station variables (MM/Bin) and one with integer station variables (MM/Int). The mathematical model using integer station variables is the first mathematical model that excludes binary assignment variables from the model, trying to reduce the size of the feasible region of the relaxed linear program. Solutions for large-sized problems which are focused in the literature are introduced first time. Experiments show that MM/Int manages to find good solutions to large-sized problems while MM/Bin fails to return a solution for most of these problems.

Despite the good performance of MM/Int, the quality of the solutions decreases as the size of the problem gets larger. In order to handle problems with greater size, 2 heuristic approaches are proposed in this study. Heuristic approaches focus on solving type-I 2SALB problems. Good solutions obtained from MM/Int give the idea of using a mathematical model in a heuristic approach. The first heuristic, called Rolling Horizon Heuristic (RHH), tries to obtain a solution by solving a modified version of MM/Int for sub-problems and proceeds like a rolling horizon. With flexible sub-problem size, the heuristic promises to use the limited resources, such as time and memory, most efficiently to find good solutions. Performance of RHH is tested with large-sized problems which are focused in the literature. RHH managed to find the optimal solutions to 19 of the 20 test problems, two of which are introduced first time.

Second heuristic proposed in this study is inspired by multiple-rule heuristic approach (Boctor, 1995). Based on numerous experiments, task selection and task assignment rules are developed to handle precedence and side constraints that result in idle times in these kinds of algorithms. Hence, the heuristic is called as Extended Multi Rule Heuristic (EMRH). Like RHH, performance of EMRH is tested with large-sized problems focused in the literature. EMRH manages to find the optimal solutions for all of the test problems, three of which are firstly introduced. Furthermore, EMRH tries to minimize the number of opened stations as a secondary objective. For 16 of the 20 problems, optimality with respect to the secondary objective is confirmed. Four of these optimal solutions are again introduced for the first time.

Heuristics introduced in this study are compared with the proposed heuristics in the literature. Experiments shows that EMRH and RHH manages to find at least as good as the solutions found by the ant-colony algorithm (ACO) introduced by Baykasoglu and Dereli (2008) and the ant-colony algorithm (2-ANTBAL) introduced by Simaria and Vilarinho (2009) in all of the problems and manages to perform better in some of these problems. On the other hand, branch-and-bound algorithm (B&B) proposed by Wu et al. (2008) performs as well as RHH and EMRH.

Heuristics proposed in this study are also compared with each other. By the quality of the solutions, the heuristics almost equally performs. On the other hand, EMRH finds the solutions within quite impressive solution times. Numerous experiments on deciding the selection and assignment rules that EMRH uses are the most important reason of the good solutions found by this heuristic. However, there is always the risk of observing problematic precedence and side constraints that cannot be handled by the rules that EMRH uses. On the other hand, RHH always promises good solutions as it exploits the advantages of mathematical modeling. The flexibility of RHH on the planning horizon size allows using limited resources, i.e. time and memory, most efficiently.

For future studies, mathematical models that use integer station variables only excluding the binary assignment variables may be modified for special types of 2SALB problems, such as problems with zoning constraints or mixed-model problems. Also, RHH and EMRH may be modified for different types of 2SALB problems.

Furthermore, EMRH may be regarded as an analysis tool for better understanding 2SALB problems. At the end of the run of the algorithm, a summary of the best solutions found by sub-algorithms, different combinations of selection and assignment rules, is provided. This report may be used to understand which combinations work best and how they may be improved.

REFERENCES

Agrawal, P. (1985). The related activity concept in assembly line balancing. *International Journal of Production Research , 23*, 403-421.

Bartholdi, J. (1993). Balancing two-sided assembly lines: A case study. *International Journal of Production Research , 31*, 2447-2461.

Bautista, J., & Pereira, J. (2006). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research , 177*, 2016-2032.

Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science , 32*, 909-932.

Baykasoglu, A., & Dereli, T. (2008). Two-sided assembly line balancing using ant-colony based heuristic. *International Journal of Advanced Manufacturing Technology , 36*, 582-588.

Boctor, F. (1995). A multiple-rule heuristic for assembly line balancing. *Journal of Operational Research Society , 55*, 589-597.

Bowman, E. (1960). Assembly line balancing by linear programming. *Operations Research , 8*, 3.

Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research , 183*, 674-693.

Bryton, B. (1954). Balancing a continuous production line. *Unpublished M.S. Thesis, Northwestern University, Evanston, ILL .*

Burns, L., & Daganzo, C. (1987). Assembly line job sequencing principles. *International Journal of Production Research , 25*, 71-99.

Dobson, G., & Yano, C. (1994). Cyclic scheduling to minimize inventory in a batch flow line. *European Journal of Operational Research , 75*, 441-461.

Gutjahr, A., & Nemhauser, G. (1964). An algorithm for the line balancing problem. *Management Science , 11*, 2.

Held, M., & Karp, R. (1962). A dynamic programming approach to sequencing problem. *Journal of the Society of Industrial and Applied Math. , 10*, 1.

Held, M., Karp, R., & Shareshian, R. (1963). Assembly line balancing - dynamic programming with precedence constraints. *Operations Research , 11*, 3.

Helgeson, W., & Birnie, D. (1961). Assembly line balancing using the ranked positional weight technique. *Journal of Industrial Engineering , 12*, 6.

Hoffman, T. (1963). Assembly line balancing with a precedence matrix. *Management Science , 9*, 4.

Hoffman, T. (1992). EUREKA: A hybrid system for assembly line balancing. *Management Science , 38*, 39-47.

Hu, T. (1961). Parallel sequencing and assembly line problems. *Operations Research , 9*.

Jackson, J. (1956). A computing procedure for a line balancing problem. *Management Science , 2*, 3.

Johnson, R. (1981). Assembly line balancing algorithms: computational comparisons. *International Journal of Production Research , 19*, 3.

Johnson, R. (1988). Optimally balancing large assembly lines with "FABLE". *Management Science , 34*, 240-253.

Kao, E., & Queyranne, M. (1982). On dynamic programming methods for assembly line balancing. *Operations Research , 30*, 2.

Kim, Y., Kim, Y., & Kim, Y. (2000). Two-sided assembly line balancing: a genetic algorithm approach. *Production Planning & Control , 11* (1), 44-53.

Kim, Y., Song, W., & Kim, J. (2009). A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers and Operations Research , 36*, 853-865.

Klein, M. (1963). On assembly line balancing. *Operations Research , 11*, 2.

Lee, T., Kim, Y., & Kim, Y. (2001). Two-sided assembly line balancing to maximize work relatedness and slackness. *Computers and Industrial Engineering , 40*, 273-292.

Miltenburg, J., & Wijngaard, J. (1994). The U-line line balancing problem. *Management Science , 40*, 1378-1388.

Ozcan, U., & Toklu, B. (2009). A tabu search algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology , 43*, 822-829.

Ozcan, U., & Toklu, B. (2009). Balancing of mixed-model two-sided assembly lines. *Computers and Indutrial Engineering , 57*, 217-227.

Ozcan, U., & Toklu, B. (2009). Multiple-criteria decision-making in two-sided assembly line balancing: programming and a fuzzy goal programming models. *Computers and Operations Research , 36*, 1955-1965.

Patterson, J., & Albracht, J. (1975). Assembly-line balancing: zero-one programming with fibonacci search. *Operations Research , 23*, 166-172.

Salveson, M. (1955). The assembly line balancing problem. *Journal of Industrial Engineering , 6*, 18-25.

Scholl, A. (1999). *Balancing and sequencing assembly lines.* Heidelberg.

Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operations Research , 168*, 666-693.

Scholl, A., & Klein, R. (1999). ULINO: Optimally balancing U-shaped JIT assembly lines. *International Journal of Production Research , 37*, 721-736.

Schrage, L., & Baker, K. (1978). Dynamic programming solution of sequencing problems with precedence constraints. *Operations Research , 26*.

Shtub, A., & Dar-El, E. (1989). A methodology for the selection of assembly systems. *International Journal of Production Research , 27*, 175-186.

Simaria, A., & Vilarinho, P. (2009). 2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines. *Computers and Industrial Engineering , 56*, 489-506.

Talbot, F., & Patterson, J. (1984). An integer programming algorithm with network cuts for solving the assembly line balancing problem. *Management Science , 3*, 1.

Thangavelu, S., & Shetty, C. (1971). Assembly line balancing by zero-one programming. *AIIE Transactions , 3*, 1.

Van Assche, F., & Herroelen, W. (1979). An optimal procedure for the single model deterministic assembly line balancing problem. *European Journal of Operational Research , 3*, 142-149.

Wee, T., & Magazine, M. (1981). An efficient branch and bound algorithm for an assembly line balancing problem-part II: maximize the production rate. Working Paper no. 151, University of Waterloo, Ontario, Canada, June.

White, W. (1961). Comments on a paper by Bowman. *Operations Research , 9*, 247-277.

Wu, E., Jin, Y., Bao, J., & Hu, X. (2008). A branch-and-bound algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology , 39*, 1009-1015.

Xiaofeng, H., Wu, E., & Ye, J. (2008). A station-oriented enumerative algorithm for two-sided assembly line balancing. *European Journal of Operational Research , 186*, 435-440.

APPENDIX A

DATA OF LARGE-SIZED PROBLEMS

Table A.1 Data of 65-Task Problem

Task No	Side	Task Time	Immediate Successors
1	E	49	3
2	E	49	3
3	E	71	4, 23
4	E	26	5, 6, 7, 9, 11, 12, 25, 26, 27, 41, 45, 49
5	E	42	14
6	E	30	14
7	R	167	8
8	R	91	14
9	L	52	10
10	L	153	14
11	E	68	14
12	E	52	14
13	E	135	14
14	E	54	15, 18, 20, 22
15	E	57	16
16	L	151	17
17	L	39	31
18	R	194	19
19	R	35	21
20	E	119	21
21	E	34	31
22	E	38	31
23	E	104	24
24	E	84	31
25	L	113	31

Table A.1 (continued)

26	R	72	31
27	R	62	28
28	R	272	50
29	L	89	50
30	L	49	50
31	E	11	32, 36, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62
32	E	45	33
33	E	54	34
34	E	106	35
35	R	132	50
36	E	52	37
37	E	157	38
38	E	109	39, 40
39	L	32	50
40	R	32	50
41	E	52	42
42	E	193	43
43	E	34	62
44	R	34	46
45	L	97	46
46	E	37	47
47	L	25	48
48	L	89	50
49	E	27	50
50	E	50	65
51	R	46	65
52	E	46	65
53	L	55	65
54	E	118	65
55	R	47	65
56	E	164	57
57	E	113	65
58	L	69	65
59	R	30	65

Table A.1 (continued)

60	E	25	65
61	R	106	65
62	E	23	63
63	L	118	64
64	L	155	65
65	E	65	-

Table A.2 Data of 205-Task Problem

Task No	**Side**	**Task Time**	**Immediate Successors**
1	E	692	36
2	E	42	3, 4
3	R	261	5
4	L	261	5
5	E	157	7, 13
6	E	90	36
7	R	54	8
8	R	67	9
9	R	30	10
10	R	106	11
11	R	32	12
12	R	62	36
13	L	54	14
14	L	67	15
15	L	30	16
16	L	106	17
17	L	32	18
18	L	62	36
19	E	56	36
20	E	67	22
21	E	86	22
22	E	37	23
23	E	41	24, 34
24	E	72	26, 27, 28

Table A.2 (continued)

25	R	86	28
26	L	16	35
27	R	51	35
28	R	66	29
29	R	41	30, 33
30	R	72	31, 32
31	R	51	35
32	R	16	35
33	R	15	35
34	L	15	35
35	E	85	36
36	E	59	37, 40, 41, 42, 62, 69, 72, 75, 83, 110, 111, 112
37	L	23	38
38	L	13	39
39	L	19	45
40	E	108	43, 54
41	E	214	92
42	E	80	43, 54
43	L	37	44
44	L	84	45
45	L	18	46, 48, 51, 53
46	L	12	47
47	L	29	92
48	L	37	49
49	L	13	50
50	L	70	92
51	L	217	52
52	L	72	92
53	L	85	92
54	R	43	55, 133
55	R	97	56, 59, 61
56	R	37	57
57	R	13	58

Table A.2 (continued)

58	R	35	92
59	R	217	60
60	R	72	92
61	R	85	92
62	E	25	63
63	E	37	64
64	E	37	65, 68
65	E	103	66
66	E	140	67
67	E	49	80
68	E	35	80
69	E	51	70
70	E	88	71
71	E	53	73
72	E	144	73
73	E	337	74
74	E	107	76
75	E	371	92
76	E	97	77, 78, 79
77	E	166	80, 82
78	L	92	80
79	R	92	80
80	E	106	81
81	E	49	84
82	E	92	92
83	E	371	92
84	E	87	85
85	E	162	86, 88, 90
86	E	96	87
87	E	79	92
88	E	96	89
89	E	42	92
90	R	88	91
91	R	90	92

Table A.2 (continued)

92	R	97	93, 94, 95, 96, 97, 98, 99
93	R	270	135
94	E	452	135
95	R	48	113
96	E	338	113
97	E	34	100
98	E	65	100
99	E	50	100
100	E	112	101, 103, 105, 109, 130, 131, 134
101	E	48	102
102	E	117	113
103	E	50	104
104	R	68	113
105	L	232	106, 107
106	L	122	108
107	E	151	108
108	L	31	113
109	E	97	113
110	R	308	113
111	L	116	113
112	R	312	113
113	E	34	114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 161, 162, 163, 169
114	L	128	160
115	E	54	160
116	R	175	160
117	E	55	160
118	E	306	126
119	E	59	126
120	E	59	126
121	E	66	126
122	E	66	126
123	E	23	126
124	E	244	125

Table A.2 (continued)

125	E	54	126
126	R	294	127, 128, 129
127	E	84	135
128	E	61	135
129	E	57	135
130	R	38	136
131	E	944	132
132	R	511	133
133	R	625	189
134	R	445	189
135	L	68	136, 137, 138, 139, 140, 141, 142, 144, 145, 147, 148, 149, 150, 151, 152
136	L	53	189
137	E	49	160
138	E	92	160
139	E	236	160
140	L	116	143
141	L	265	143
142	L	149	143
143	L	74	160
144	E	332	160
145	E	324	146
146	L	104	160
147	L	51	160
148	R	58	160
149	R	67	160
150	R	49	160
151	E	107	160
152	L	38	160
153	L	27	154
154	E	68	155
155	E	207	156
156	E	202	157
157	E	83	189

Table A.2 (continued)

158	R	35	159
159	R	58	189
160	E	42	164, 170, 178, 179, 184
161	R	68	167
162	R	68	165
163	R	68	164
164	R	103	165
165	R	103	166
166	R	103	167
167	R	103	168
168	R	103	177
169	L	68	170
170	L	103	172
171	L	68	172
172	L	103	173
173	L	103	175
174	L	68	175
175	L	103	176
176	L	103	177
177	E	10	185, 186, 187, 188, 194, 195
178	E	187	180
179	L	134	180
180	L	89	181, 183
181	L	58	182
182	L	49	-
183	L	134	-
184	L	53	-
185	E	334	189
186	R	24	189
187	R	76	189
188	L	76	189
189	E	192	190, 191, 193
190	E	98	-
191	R	258	192

Table A.2 (continued)

192	E	165	-
193	R	38	-
194	E	115	197
195	L	83	196
196	R	56	197
197	R	29	198, 199
198	R	303	-
199	R	18	-
200	R	29	-
201	L	154	-
202	L	90	-
203	L	93	-
204	E	94	-
205	E	165	-

Table A.3 Data of 148-Task Problem

Task No	Side	Task Time	Immediate Predecessors
1	E	16	
2	E	30	
3	E	7	2
4	E	47	3
5	E	29	1, 3
6	E	8	1, 3
7	E	39	1, 3
8	E	37	1, 4
9	E	32	6
10	E	29	8
11	E	17	
12	E	11	11
13	E	32	12
14	E	15	5, 7, 9, 10
15	L	53	14
16	R	53	14
17	E	8	15, 16
18	L	24	17
19	R	24	17

Table A.3 (continued)

20	E	8	18, 19
21	R	7	20
22	L	8	20
23	L	14	20
24	R	13	20
25	R	10	21, 22, 23, 24
26	R	25	21, 22, 23, 24
27	L	11	21, 22, 23, 24
28	L	25	21, 22, 23, 24
29	E	11	25, 26, 27, 28
30	R	29	
31	E	25	29
32	L	10	
33	R	14	
34	L	41	32
35	R	42	33
36	R	47	31, 34, 35
37	R	7	36
38	R	80	37
39	R	7	38
40	R	41	39
41	R	47	40
42	L	16	
43	L	32	42
44	L	66	43
45	L	80	37
46	L	7	45
47	L	41	46
48	E	13	40, 47
49	L	47	47
50	E	33	
51	L	34	50
52	L	11	
53	L	118	51, 52
54	L	25	40, 47
55	R	7	54
56	E	28	
57	L	12	
58	L	52	
59	E	14	

Table A.3 (continued)

60	E	3	
61	E	3	
62	E	8	61
63	E	16	62
64	R	33	
65	E	8	64
66	E	18	65
67	E	10	63, 66
68	E	14	67
69	R	28	51
70	R	11	
71	R	118	64, 70
72	R	25	54, 64
73	E	40	56
74	E	40	
75	E	101	59, 74
76	E	5	54
77	E	28	76
78	E	8	77
79	E	111	69
80	E	7	79
81	E	26	80
82	E	10	57, 78, 81
83	E	21	82
84	E	26	82
85	E	20	
86	E	21	58, 73
87	E	47	86
88	E	23	58, 73
89	E	13	54, 59, 73
90	E	19	54, 73, 75
91	E	115	
92	E	35	
93	L	26	
94	E	46	
95	E	20	68
96	E	31	73
97	E	19	75
98	E	34	68
99	E	51	65

Table A.3 (continued)

100	E	39	99
101	E	30	95, 98, 100
102	E	26	101
103	E	13	101
104	E	45	96
105	E	58	91
106	E	28	84
107	E	8	106
108	E	43	107
109	E	40	108
110	E	34	109
111	E	23	90
112	L	162	111
113	L	11	112
114	E	19	113
115	E	14	114
116	E	31	113
117	E	32	116
118	E	26	117
119	E	55	105
120	E	31	113
121	E	32	120
122	E	26	121
123	E	19	113
124	E	14	123
125	E	19	115, 124
126	E	48	118, 122
127	E	55	102, 103
128	L	8	113
129	L	11	128
130	L	27	129
131	L	18	130
132	E	36	
133	L	23	54, 55
134	R	20	72
135	E	46	92, 132, 133, 134
136	E	64	135
137	L	22	130
138	E	15	
139	E	34	138

Table A.3 (continued)

140	E	22	139
141	L	151	
142	R	148	141
143	L	64	142
144	L	170	
145	R	137	144
146	R	64	142
147	L	78	142, 145
148	R	78	142, 145

APPENDIX B

MATHEMATICAL MODEL SOLUTIONS FOR TYPE-I PROBLEMS

Table B.1 MM/Int-I Solution for 65-Task Problem with CT=381 and CT=490

	CT = 381			CT = 490		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	L	49	1	L	49
2	1	R	87	1	L	98
3	1	R	158	1	R	206
4	1	R	184	1	L	240
5	1	L	346	2	L	94
6	2	L	191	1	L	390
7	2	R	290	2	R	204
8	2	R	381	2	R	295
9	2	L	52	2	L	337
10	2	L	381	2	L	490
11	1	L	304	1	L	308
12	1	L	236	1	L	360
13	1	L	184	1	R	135
14	3	L	54	3	L	54
15	3	R	373	3	L	111
16	4	L	264	3	L	381
17	4	L	337	3	L	420
18	3	R	316	3	R	300
19	4	R	35	3	R	373
20	3	L	366	3	L	230
21	4	L	298	3	R	407
22	3	R	122	3	R	338
23	2	R	104	1	R	406
24	3	R	84	1	R	490
25	4	L	113	2	L	232
26	1	R	256	3	R	72

Table B.1 (continued)

27	1	R	370	1	R	302
28	4	R	341	4	R	459
29	6	L	112	5	L	112
30	5	L	376	1	L	147
31	4	L	348	3	L	431
32	5	L	163	4	L	163
33	6	L	166	4	L	217
34	6	R	272	4	L	323
35	7	R	257	6	R	212
36	5	R	52	3	L	490
37	5	R	372	4	R	157
38	6	R	381	5	L	339
39	7	L	150	6	L	212
40	7	R	125	5	R	371
41	1	R	308	2	L	52
42	3	L	247	2	R	488
43	4	R	69	3	R	106
44	1	R	34	1	R	240
45	2	L	149	1	L	490
46	2	L	228	2	R	37
47	4	L	373	2	L	119
48	6	L	324	5	L	483
49	5	R	215	3	R	434
50	7	R	307	6	R	262
51	7	R	46	3	R	480
52	6	R	46	6	R	308
53	6	L	379	5	L	394
54	5	L	118	4	L	118
55	7	R	93	6	R	47
56	5	L	327	4	L	487
57	6	R	159	6	R	421
58	6	L	235	6	L	421
59	5	R	188	4	R	187
60	4	R	373	6	L	25
61	5	R	158	5	R	477
62	6	L	23	5	L	23
63	7	L	118	5	L	230
64	7	L	307	6	L	180
65	7	R	372	6	R	486

Table B.2 MM/Int-I Solution to 65-Task Problem with CT=544

	CT = 544		
Task	**Station**	**Side**	**Finish Time**
1	1	R	83
2	1	R	132
3	1	R	203
4	1	R	229
5	2	L	139
6	1	L	379
7	1	R	396
8	2	R	304
9	1	L	544
10	2	L	408
11	1	L	349
12	1	L	281
13	1	L	135
14	2	L	462
15	2	R	519
16	3	L	270
17	3	L	309
18	3	R	194
19	3	R	267
20	3	L	119
21	3	R	301
22	3	R	232
23	2	R	104
24	2	R	388
25	1	L	492
26	2	R	213
27	2	R	450
28	4	R	537
29	1	L	224
30	2	L	511
31	3	L	320
32	3	L	365
33	3	L	419
34	4	L	133
35	4	R	265
36	4	L	387
37	4	L	544

Table B.2 (continued)

35	4	R	265
36	4	L	387
37	4	L	544
38	5	R	109
39	5	L	187
40	5	R	141
41	1	R	475
42	3	R	494
43	4	R	110
44	1	R	34
45	2	L	97
46	2	R	141
47	2	L	166
48	2	L	255
49	1	R	423
50	5	L	474
51	4	R	76
52	4	L	179
53	5	L	242
54	3	L	537
55	3	R	541
56	5	R	311
57	5	L	424
58	5	L	311
59	4	R	30
60	4	L	25
61	5	R	474
62	4	R	133
63	4	L	297
64	5	L	155
65	5	R	539

APPENDIX C

MATHEMATICAL MODEL SOLUTIONS FOR TYPE-II PROBLEMS

Table C.1 MM/Int-II Solution to 65-Task Problem with N=4 and N=5

	N = 4			N = 5		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	R	49	1	L	49
2	1	L	49	1	R	49
3	1	R	154	1	L	120
4	1	R	180	1	L	146
5	1	L	482	1	L	188
6	2	L	211	2	R	155
7	1	R	347	1	R	537
8	2	R	91	2	R	125
9	1	L	440	1	L	360
10	1	L	652	1	L	537
11	2	L	68	1	L	256
12	1	R	652	1	L	308
13	1	R	482	1	R	184
14	2	L	265	2	R	209
15	2	R	450	2	L	347
16	2	L	619	2	L	498
17	3	L	73	2	L	537
18	2	R	652	2	R	507
19	3	R	35	3	R	76
20	2	L	468	3	L	257
21	3	R	69	3	L	291
22	3	R	134	3	R	114
23	1	L	388	2	R	313
24	2	L	349	3	L	375
25	2	L	181	3	L	113
26	2	R	200	1	R	318

Table C.1 (continued)

27	1	R	544	1	R	246
28	3	R	464	3	R	386
29	1	L	138	3	L	537
30	1	L	187	4	L	49
31	3	R	145	3	L	386
32	3	L	207	3	R	431
33	3	L	261	3	R	537
34	3	L	367	4	L	533
35	4	R	392	5	R	292
36	3	L	419	4	R	52
37	3	R	621	4	L	240
38	4	L	164	5	L	301
39	4	L	432	5	L	422
40	4	R	230	5	R	370
41	1	R	596	1	R	370
42	2	R	393	2	L	193
43	3	L	34	4	L	83
44	1	R	83	2	R	34
45	1	L	284	2	L	290
46	2	R	128	3	R	37
47	2	L	652	3	L	138
48	3	L	162	5	L	390
49	3	R	96	2	R	537
50	4	R	555	5	L	472
51	4	R	92	5	R	338
52	4	R	46	3	R	477
53	4	L	55	3	L	441
54	4	L	400	4	R	494
55	3	R	192	5	R	160
56	3	L	652	4	R	270
57	4	R	505	5	R	113
58	3	L	488	4	L	427
59	4	R	260	4	R	82
60	4	R	587	5	L	180
61	4	R	198	4	R	376
62	3	R	652	4	R	106
63	4	L	282	4	L	358
64	4	L	587	5	L	155
65	4	R	652	5	R	537

Table C.2 MM/Int-II Solutions to 65-Task Problem with N=6 and N=7

	N = 6			N = 7		
Task	Station	Side	Finish Time	Station	Side	Finish Time
1	1	R	83	1	R	49
2	1	L	49	1	L	49
3	1	R	154	1	L	120
4	1	R	180	1	R	210
5	1	L	330	1	L	374
6	2	L	134	2	L	272
7	1	R	426	2	R	194
8	2	R	91	2	R	374
9	2	L	52	1	L	276
10	2	L	287	2	L	242
11	2	R	352	1	R	312
12	2	L	104	1	L	328
13	1	L	184	1	R	184
14	2	L	438	3	L	54
15	3	L	94	3	R	141
16	3	L	364	3	L	324
17	3	L	441	4	L	39
18	3	R	228	3	R	374
19	3	R	263	4	R	35
20	3	L	213	3	L	173
21	3	R	443	4	L	73
22	3	L	402	3	R	179
23	1	L	288	1	L	224
24	3	R	409	3	R	84
25	1	L	443	4	L	186
26	2	R	424	2	R	266
27	3	R	325	1	R	374
28	5	R	325	4	R	307
29	4	L	163	2	L	89
30	4	L	49	3	L	373
31	4	R	11	4	L	197
32	4	L	233	4	L	242
33	4	L	287	4	R	374
34	5	L	106	6	L	106
35	6	R	277	6	R	238
36	4	R	63	5	R	52
37	4	R	313	5	L	209
38	5	R	434	5	L	318

Table C.2 (continued)

38	5	R	434	5	L	318
39	6	L	150	6	L	138
40	6	R	32	7	R	196
41	1	R	259	5	L	52
42	2	R	284	5	R	245
43	3	R	34	6	R	34
44	1	R	34	1	R	244
45	2	L	384	2	L	374
46	3	L	37	5	R	282
47	4	L	188	6	L	186
48	4	L	431	7	L	89
49	1	R	207	2	R	27
50	6	R	327	7	L	309
51	4	R	109	5	R	328
52	5	L	152	5	R	374
53	4	L	342	5	L	374
54	6	L	118	4	L	374
55	4	R	156	6	R	106
56	5	L	316	7	R	164
57	6	R	145	7	R	309
58	6	L	374	6	L	374
59	5	R	53	6	R	268
60	4	L	74	6	R	59
61	4	R	419	6	R	374
62	5	R	23	6	L	161
63	5	L	434	6	L	304
64	6	L	305	7	L	244
65	6	R	439	7	R	374

Table C.3 MM/Int-II Solution to 65-Task Problem with N=8

	N = 8		
Task	**Station**	**Side**	**Finish Time**
1	1	L	49
2	1	R	49
3	1	R	120
4	1	R	250
5	1	R	292
6	3	L	30
7	2	R	167
8	2	R	330
9	1	L	325
10	2	L	153
11	2	L	221
12	2	L	273
13	1	L	273
14	3	L	84
15	3	R	141
16	3	L	319
17	4	L	158
18	4	R	290
19	4	R	325
20	4	L	119
21	5	R	34
22	4	L	196
23	1	R	224
24	3	L	168
25	4	L	309
26	2	R	239
27	4	R	96
28	6	R	325
29	1	L	138
30	6	L	336
31	5	R	45
32	5	R	90
33	5	R	144
34	7	L	106
35	8	R	210
36	5	L	174
37	5	L	331
38	7	R	109

Table C.3 (continued)

39	8	L	150
40	8	R	32
41	2	L	325
42	3	R	334
43	4	R	34
44	1	R	326
45	5	L	97
46	5	R	181
47	6	L	25
48	6	L	114
49	3	R	27
50	8	R	260
51	5	R	333
52	8	R	78
53	6	L	287
54	8	L	118
55	7	R	156
56	7	R	320
57	8	L	263
58	7	L	330
59	6	R	30
60	5	L	122
61	5	R	287
62	6	R	53
63	6	L	232
64	7	L	261
65	8	R	328

APPENDIX D

EMRH SOLUTIONS FOR TYPE-I PROBLEMS

Table D.1 EMRH Solutions to 65-Task Problem with CT=381 and CT=490

	CT = 381			CT = 490		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	R	49	1	R	49
2	1	L	49	1	L	49
3	1	L	120	1	R	120
4	1	L	146	1	R	146
5	2	R	253	1	R	188
6	1	L	381	1	L	214
7	1	R	351	1	R	355
8	2	R	91	1	R	446
9	1	L	198	1	L	266
10	1	L	351	1	L	419
11	2	R	159	2	R	68
12	2	R	211	2	L	52
13	1	R	184	1	L	184
14	2	R	307	2	R	122
15	3	L	57	2	R	179
16	3	L	208	2	L	426
17	3	L	247	3	L	39
18	3	R	194	2	R	373
19	3	R	348	2	R	408
20	3	R	313	2	L	275
21	4	L	34	2	R	442
22	2	L	379	4	L	485
23	2	L	104	2	L	156
24	4	R	84	3	R	84
25	2	L	217	3	L	152
26	2	L	289	3	L	224
27	5	L	203	2	L	488
28	6	R	272	3	R	356
29	7	L	89	3	L	313

Table D.1 (continued)

30	7	L	227	3	L	362
31	4	R	95	3	R	367
32	5	R	154	3	R	412
33	5	R	208	4	R	54
34	5	R	314	4	R	160
35	7	R	132	4	R	292
36	4	R	147	4	L	52
37	4	L	355	4	L	209
38	5	R	109	4	L	318
39	6	L	369	4	L	350
40	5	R	346	4	R	442
41	2	L	341	3	L	414
42	4	R	340	5	R	193
43	4	R	374	5	R	227
44	2	R	341	1	R	480
45	3	L	344	4	L	447
46	3	L	381	4	R	484
47	4	L	380	5	L	25
48	7	L	178	5	L	114
49	2	R	368	1	L	446
50	7	L	277	5	L	475
51	6	R	365	3	R	458
52	7	R	284	3	L	460
53	5	L	376	5	L	169
54	5	L	321	4	R	410
55	6	R	319	2	R	489
56	4	L	198	5	L	333
57	6	L	268	5	R	446
58	6	L	337	5	L	402
59	1	R	381	3	R	488
60	3	R	373	1	L	471
61	7	R	238	5	R	333
62	5	L	23	5	L	425
63	5	L	141	6	L	118
64	6	L	155	6	L	273
65	7	L	349	6	L	338

Table D.2 EMRH Solution to 65-Task Problem with CT=544

	CT = 544		
Task	**Station**	**Side**	**Finish Time**
1	1	R	49
2	1	L	49
3	1	R	120
4	1	R	146
5	1	L	278
6	1	L	308
7	1	R	313
8	1	R	404
9	1	L	236
10	1	L	461
11	1	R	472
12	1	L	513
13	1	L	184
14	2	R	54
15	2	R	111
16	2	L	374
17	2	L	413
18	2	R	305
19	2	R	340
20	2	L	223
21	2	R	374
22	3	L	542
23	2	L	104
24	2	R	458
25	3	L	113
26	2	L	485
27	3	L	504
28	4	R	436
29	4	L	366
30	4	L	415
31	3	L	124
32	3	R	238
33	3	R	326
34	3	R	455
35	5	R	132
36	3	L	176
37	3	L	333
38	3	L	442

Table D.2 (continued)

39	4	L	447
40	3	R	487
41	2	R	510
42	3	R	193
43	3	R	272
44	1	R	506
45	4	L	97
46	4	L	134
47	4	L	159
48	4	L	536
49	1	L	540
50	5	R	288
51	4	R	482
52	4	R	528
53	2	L	540
54	5	L	118
55	3	R	534
56	4	R	164
57	5	L	231
58	5	L	300
59	1	R	536
60	2	R	535
61	5	R	238
62	3	R	349
63	4	L	277
64	5	L	455
65	5	L	520

Table D.3 EMRH Solutions to 205-Task Problem with CT=1133 and CT=1322

	CT = 1133			CT = 1322		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	L	1093	1	L	1149
2	1	L	109	1	L	109
3	1	R	690	1	R	690
4	1	L	370	1	L	370
5	1	R	847	1	R	847
6	2	R	330	2	R	141
7	1	R	973	1	R	973
8	1	R	1040	1	R	1040
9	1	R	1070	1	R	1070
10	2	R	106	1	R	1176
11	2	R	138	1	R	1208
12	2	R	392	2	L	200
13	2	L	54	1	L	1203
14	2	L	121	1	L	1270
15	2	L	151	1	L	1300
16	2	L	257	2	L	106
17	2	L	289	2	L	138
18	2	L	351	2	L	262
19	2	L	407	1	L	426
20	1	L	67	1	L	67
21	1	R	86	1	R	86
22	1	R	123	1	R	123
23	1	R	164	1	R	164
24	1	R	236	1	R	236
25	1	R	322	1	R	322
26	1	L	386	1	L	442
27	2	R	189	1	R	1259
28	1	R	388	1	R	388
29	1	R	429	1	R	429
30	1	R	919	1	R	919
31	2	R	240	2	R	51
32	1	R	1086	1	R	1275
33	1	R	1101	1	R	1290
34	1	L	401	1	L	457
35	2	L	492	2	R	226
36	2	L	551	2	L	321
37	2	L	1103	2	L	1221
38	2	L	1116	2	L	1234

Table D.3 (continued)

39	3	L	426	2	L	1290
40	2	R	803	2	R	741
41	3	R	902	4	L	431
42	2	R	883	2	R	821
43	3	L	407	2	L	1271
44	3	L	510	3	L	250
45	3	L	528	3	L	268
46	3	L	1073	3	L	1076
47	3	L	1115	3	L	1214
48	3	L	1061	3	L	1064
49	3	L	1086	3	L	1089
50	4	L	287	3	L	1284
51	4	L	217	4	L	217
52	4	L	359	4	L	503
53	4	L	444	4	L	588
54	3	R	232	2	R	1308
55	3	R	421	3	R	189
56	3	R	458	4	R	37
57	3	R	471	4	R	50
58	3	R	937	4	R	302
59	3	R	688	4	R	267
60	3	R	1009	4	R	374
61	3	R	1094	4	R	459
62	2	R	908	2	R	864
63	2	R	945	2	R	901
64	2	R	982	2	R	938
65	2	R	1085	2	R	1041
66	3	R	140	2	R	1181
67	3	R	189	2	R	1230
68	2	R	1120	2	R	1265
69	2	L	602	2	L	372
70	2	L	690	2	L	460
71	2	L	743	2	L	513
72	2	R	695	2	L	657
73	2	L	1080	2	L	994
74	3	L	107	2	L	1101
75	4	R	651	4	R	830
76	3	L	204	2	L	1198
77	3	L	370	3	L	166
78	3	L	620	3	L	360
79	3	R	324	3	R	92

Table D.3 (continued)

80	3	L	726	3	L	466
81	3	L	775	3	L	515
82	4	L	536	4	L	680
83	4	L	907	4	L	1051
84	3	L	862	3	L	602
85	3	L	1024	3	L	764
86	4	R	96	3	L	1185
87	4	R	730	4	R	909
88	4	R	192	3	L	985
89	4	R	772	3	L	1027
90	4	R	280	4	R	997
91	4	R	862	3	R	1302
92	4	R	1004	5	R	97
93	7	R	270	4	R	1267
94	5	L	1114	8	L	581
95	5	R	48	8	R	1260
96	5	R	386	6	R	989
97	4	R	1038	6	L	737
98	4	R	1103	6	L	802
99	4	L	1073	4	R	1317
100	5	L	112	6	R	1101
101	5	L	392	8	L	629
102	5	L	631	6	R	1218
103	5	R	436	7	R	1319
104	5	R	655	6	R	1286
105	5	L	344	7	L	1176
106	5	L	514	7	L	1298
107	5	R	587	8	L	780
108	5	L	662	8	L	811
109	5	R	752	8	L	908
110	5	R	1060	9	R	308
111	4	L	1023	8	L	1024
112	6	R	312	2	R	633
113	6	R	346	9	R	342
114	7	L	182	5	L	128
115	7	L	236	4	L	1250
116	7	R	880	5	R	272
117	6	L	1124	4	L	1305
118	6	R	896	3	R	739
119	6	R	955	3	R	798
120	6	L	1003	3	L	823

Table D.3 (continued)

121	6	R	1021	3	R	864
122	6	L	1069	3	L	889
123	6	R	1044	2	L	1313
124	6	R	590	3	R	433
125	7	L	54	3	R	918
126	7	R	564	3	R	1212
127	7	R	648	4	L	1135
128	7	L	688	4	L	1196
129	7	R	705	8	L	1268
130	5	R	1098	7	R	38
131	6	L	944	7	L	944
132	9	R	923	8	R	511
133	10	R	625	8	R	1136
134	10	R	1070	7	R	741
135	7	L	921	5	L	196
136	8	L	1122	8	L	53
137	7	R	997	5	R	645
138	7	R	1089	5	R	737
139	8	R	560	5	L	962
140	7	L	1037	5	L	312
141	8	L	265	5	L	577
142	8	L	414	5	L	726
143	8	L	488	5	L	1036
144	8	L	820	5	R	1069
145	8	R	324	5	R	596
146	8	L	924	5	L	1140
147	7	L	1088	5	L	1191
148	8	R	618	5	R	1127
149	8	R	685	5	R	1194
150	8	R	734	5	R	1243
151	8	R	841	5	L	1298
152	7	L	1126	3	L	1322
153	9	L	449	6	L	829
154	9	L	517	6	L	897
155	9	L	724	6	L	1104
156	9	L	1009	7	R	240
157	9	L	1092	7	R	824
158	7	R	1124	6	R	1321
159	8	R	967	7	R	882
160	8	L	966	6	L	42
161	8	R	909	6	R	342

Table D.3 (continued)

162	7	R	948	6	R	68
163	6	R	1112	5	R	1311
164	8	R	1070	6	R	171
165	9	R	103	6	R	274
166	9	R	206	6	R	445
167	9	R	309	6	R	548
168	9	R	412	6	R	651
169	7	L	304	6	L	110
170	8	L	1069	6	L	213
171	7	L	372	6	L	281
172	9	L	103	6	L	384
173	9	L	206	6	L	487
174	7	L	440	9	L	695
175	9	L	309	6	L	590
176	9	L	412	6	L	693
177	9	L	422	6	L	703
178	10	L	597	8	L	1211
179	10	L	731	9	L	134
180	10	L	820	9	L	415
181	10	L	878	9	L	473
182	10	L	1081	9	L	744
183	11	L	134	9	L	878
184	11	L	187	8	L	1321
185	10	L	334	7	R	1216
186	10	R	1094	7	R	1240
187	11	R	76	8	R	1212
188	10	L	410	8	L	129
189	11	R	268	9	L	326
190	11	L	375	9	R	698
191	11	R	526	9	R	600
192	11	R	691	9	R	863
193	11	R	729	9	R	901
194	9	R	1038	6	L	1302
195	9	L	807	6	L	1187
196	9	R	1094	7	R	296
197	9	R	1123	7	R	1269
198	11	R	1032	9	R	1204
199	10	R	1112	8	R	1278
200	11	R	1061	8	R	1307
201	10	L	1032	9	L	627
202	11	L	277	9	L	968

Table D.3 (continued)

203	7	L	533	9	L	1061
204	7	L	627	9	L	1155
205	7	L	853	9	L	1320

Table D.4 EMRH Solutions to 205-Task Problem with CT=1510 and CT=1699

	CT = 1510			**CT = 1699**		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	L	1149	1	L	1149
2	1	L	109	1	L	109
3	1	R	690	1	R	690
4	1	L	370	1	L	370
5	1	R	847	1	R	847
6	1	R	1431	1	R	1431
7	1	R	973	1	R	973
8	1	R	1040	1	R	1040
9	1	R	1070	1	R	1070
10	1	R	1176	1	R	1176
11	1	R	1208	1	R	1208
12	1	L	1500	1	L	1500
13	1	L	1203	1	L	1203
14	1	L	1270	1	L	1270
15	1	L	1300	1	L	1300
16	1	L	1406	1	L	1406
17	1	L	1438	1	L	1438
18	2	L	62	1	L	1562
19	1	L	426	1	L	426
20	1	L	67	1	L	67
21	1	R	86	1	R	86
22	1	R	123	1	R	123
23	1	R	164	1	R	164
24	1	R	236	1	R	236
25	1	R	322	1	R	322
26	1	L	442	1	L	442
27	1	R	1259	1	R	1259
28	1	R	388	1	R	388
29	1	R	429	1	R	429
30	1	R	919	1	R	919
31	1	R	1310	1	R	1310
32	1	R	1326	1	R	1326
33	1	R	1341	1	R	1341

Table D.4 (continued)

34	1	L	457	1	L	457
35	2	R	85	1	R	1516
36	2	R	144	1	L	1621
37	2	L	1217	1	L	1695
38	2	L	1230	2	L	1005
39	2	L	1286	2	L	1061
40	2	R	564	2	R	420
41	3	L	1209	3	L	798
42	2	R	644	2	R	500
43	2	L	1267	2	L	1042
44	2	L	1370	2	L	1145
45	2	L	1388	2	L	1163
46	2	L	1492	2	L	1694
47	3	L	1238	3	L	827
48	3	L	627	3	L	216
49	3	L	640	3	L	229
50	3	L	1308	3	L	897
51	3	L	857	3	L	488
52	3	L	1380	3	L	969
53	3	L	1465	3	L	1054
54	2	R	1223	2	R	1036
55	2	R	1320	2	R	1133
56	2	R	1498	3	R	642
57	3	R	1043	3	R	655
58	3	R	1295	3	R	907
59	3	R	1260	3	R	872
60	3	R	1367	3	R	979
61	3	R	1452	3	R	1064
62	2	R	687	1	R	1664
63	2	R	724	2	R	537
64	2	R	761	2	R	574
65	2	R	864	2	R	677
66	2	R	1004	2	R	817
67	2	R	1053	2	R	866
68	2	R	1088	2	R	901
69	2	L	202	1	L	1672
70	2	L	290	2	L	88
71	2	L	343	2	L	141
72	2	L	487	2	L	285
73	2	L	824	2	L	622
74	2	L	931	2	L	729

Table D.4 (continued)

75	4	L	371	3	L	1425
76	2	L	1028	2	L	826
77	2	L	1194	2	L	992
78	2	L	1480	2	L	1255
79	2	R	1180	2	R	993
80	3	L	106	2	L	1361
81	3	L	155	2	L	1410
82	4	R	92	3	R	1156
83	4	R	463	3	R	1527
84	3	L	242	2	L	1497
85	3	L	404	2	L	1659
86	3	L	995	3	L	584
87	4	L	450	3	L	1504
88	3	R	646	3	R	221
89	3	L	899	3	L	271
90	4	R	551	3	R	1615
91	3	R	1030	3	R	605
92	4	R	648	4	R	97
93	4	R	918	4	R	367
94	7	L	786	6	L	1467
95	6	R	1362	7	R	124
96	6	L	338	5	L	853
97	3	L	1499	3	R	1699
98	5	L	1281	4	L	1639
99	5	R	737	3	R	1665
100	6	L	450	5	L	965
101	6	R	1410	6	L	1515
102	6	L	567	5	L	1082
103	6	L	1456	6	L	1565
104	6	R	1012	5	R	1605
105	6	L	1208	6	L	347
106	6	L	1330	6	L	469
107	7	L	937	7	L	151
108	7	L	968	6	L	1596
109	7	L	1065	7	R	221
110	7	R	1444	7	R	529
111	7	L	1181	7	L	267
112	2	R	456	2	R	312
113	7	L	1478	7	L	622
114	4	L	723	4	L	128
115	4	L	777	4	L	182

Table D.4 (continued)

116	4	R	1093	4	R	542
117	4	L	832	4	L	237
118	3	R	550	2	R	1683
119	2	R	1379	3	L	59
120	2	R	1438	3	R	59
121	3	L	470	3	L	125
122	3	L	536	3	R	125
123	2	R	1461	2	L	1682
124	3	R	244	2	R	1377
125	3	L	590	3	L	179
126	3	R	940	3	R	515
127	4	L	534	3	L	1588
128	4	L	595	3	L	1649
129	3	R	1509	4	L	1696
130	5	R	1368	4	R	1698
131	6	R	944	5	R	1537
132	7	R	511	6	R	511
133	7	R	1136	6	R	1192
134	8	R	445	6	R	1637
135	4	L	1036	4	L	492
136	5	L	1485	6	L	522
137	4	L	1466	4	R	915
138	5	R	92	4	R	1007
139	5	R	328	4	R	1243
140	4	L	1152	4	L	608
141	4	L	1417	4	L	873
142	5	L	149	4	L	1022
143	5	L	223	4	L	1096
144	5	L	555	4	L	1428
145	4	R	1417	4	R	866
146	5	L	659	4	L	1532
147	2	L	113	4	L	288
148	4	R	1475	4	R	1301
149	5	R	395	4	R	1368
150	5	R	444	4	R	1417
151	5	R	551	4	R	1524
152	2	L	151	3	L	1687
153	4	L	1493	5	L	1109
154	5	L	1349	5	L	1177
155	6	L	774	5	L	1384
156	6	L	976	5	L	1669

Table D.4 (continued)

157	6	R	1151	6	L	605
158	4	R	1510	5	R	1640
159	6	R	1209	5	R	1698
160	5	L	701	4	L	1574
161	5	R	1011	5	R	274
162	5	R	687	4	R	1660
163	5	R	619	4	R	1592
164	5	R	840	5	R	103
165	5	R	943	5	R	206
166	5	R	1114	5	R	377
167	5	R	1217	5	R	480
168	5	R	1320	5	R	583
169	4	L	900	4	L	356
170	5	L	804	5	L	103
171	4	L	968	4	L	424
172	5	L	907	5	L	206
173	5	L	1010	5	L	309
174	7	L	1436	6	L	1664
175	5	L	1113	5	L	412
176	5	L	1216	5	L	515
177	5	R	1330	5	R	593
178	7	L	1368	7	L	454
179	8	L	134	7	L	588
180	8	L	223	7	L	711
181	8	L	281	7	L	769
182	8	L	484	7	L	972
183	8	L	618	7	L	1106
184	6	L	1509	7	L	1159
185	7	L	334	6	L	939
186	5	R	1507	6	R	1661
187	6	R	1285	7	R	76
188	6	L	1406	6	L	1015
189	8	R	637	7	R	721
190	8	L	806	7	R	1095
191	8	R	895	7	R	979
192	8	R	1060	7	R	1260
193	8	R	1098	7	R	1298
194	5	R	1483	6	L	115
195	5	L	1432	5	L	1467
196	6	R	1068	6	R	567
197	6	R	1314	6	R	1690

Table D.4 (continued)

198	8	R	1401	7	R	1601
199	6	R	1428	7	R	997
200	6	R	1457	7	R	1630
201	8	L	435	7	L	923
202	8	L	708	7	L	1249
203	8	L	899	7	L	1342
204	8	L	993	7	L	1436
205	8	L	1158	7	L	1601

Table D.5 EMRH Solutions to 205-Task Problem with CT=1888 and CT=2077

	CT = 1888			**CT = 2077**		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	L	1093	1	L	1093
2	1	L	109	1	L	109
3	1	R	690	1	R	690
4	1	L	370	1	L	370
5	1	R	847	1	R	847
6	1	R	1431	1	R	1431
7	1	R	973	1	R	973
8	1	R	1040	1	R	1040
9	1	R	1070	1	R	1070
10	1	R	1176	1	R	1176
11	1	R	1208	1	R	1208
12	1	R	1493	1	R	1493
13	1	L	1147	1	L	1147
14	1	L	1214	1	L	1214
15	1	L	1244	1	L	1244
16	1	L	1350	1	L	1350
17	1	L	1382	1	L	1382
18	1	L	1444	1	L	1444
19	1	L	1500	1	L	1500
20	1	L	67	1	L	67
21	1	R	86	1	R	86
22	1	R	123	1	R	123
23	1	R	164	1	R	164
24	1	R	236	1	R	236
25	1	R	322	1	R	322
26	1	L	386	1	L	386
27	1	R	1259	1	R	1259

Table D.5 (continued)

28	1	R	388	1	R	388
29	1	R	429	1	R	429
30	1	R	919	1	R	919
31	1	R	1310	1	R	1310
32	1	R	1326	1	R	1326
33	1	R	1341	1	R	1341
34	1	L	401	1	L	401
35	1	L	1585	1	L	1585
36	1	L	1644	1	L	1644
37	1	L	1859	1	L	1939
38	1	L	1872	1	L	1952
39	2	L	763	1	L	2008
40	2	R	108	1	R	1896
41	2	R	1228	2	R	962
42	2	R	188	1	L	1916
43	2	L	744	1	L	1989
44	2	L	847	2	L	791
45	2	L	865	2	L	809
46	2	L	1410	2	L	1354
47	2	L	1669	2	L	1709
48	2	L	1398	2	L	1342
49	2	L	1423	2	L	1367
50	2	L	1739	2	L	1779
51	2	L	1640	2	L	1584
52	2	L	1811	2	L	1851
53	3	L	85	2	L	1936
54	2	R	558	1	R	2073
55	2	R	747	2	R	389
56	2	R	784	2	R	426
57	2	R	797	2	R	439
58	2	R	1263	2	R	997
59	2	R	1014	2	R	656
60	2	R	1335	2	R	1069
61	2	R	1420	2	R	1154
62	1	R	1813	1	R	1921
63	1	R	1850	1	R	1958
64	1	R	1887	1	R	1995
65	2	R	291	2	R	103
66	2	R	431	2	R	243
67	2	R	480	2	R	292
68	2	R	515	1	R	2030

Table D.5 (continued)

69	1	L	1695	1	L	1695
70	1	L	1783	1	L	1783
71	1	L	1836	1	L	1836
72	1	R	1788	1	R	1788
73	2	L	337	2	L	337
74	2	L	444	2	L	444
75	3	R	371	2	R	1525
76	2	L	541	2	L	541
77	2	L	707	2	L	707
78	2	L	957	2	L	901
79	2	R	650	2	R	748
80	2	L	1063	2	L	1007
81	2	L	1112	2	L	1056
82	2	R	1792	2	R	1801
83	3	L	456	3	L	371
84	2	L	1199	2	L	1143
85	2	L	1361	2	L	1305
86	2	R	1516	2	R	1621
87	3	R	450	2	R	1880
88	2	R	1612	2	L	1680
89	2	L	1853	2	R	1922
90	2	R	1700	2	R	1709
91	2	R	1882	2	R	2012
92	3	R	855	3	R	717
93	4	R	642	3	R	1611
94	4	L	839	4	R	518
95	3	R	1197	3	R	1059
96	3	L	1258	3	L	1120
97	3	R	889	3	R	751
98	3	L	920	3	L	782
99	3	R	939	3	R	801
100	3	R	1051	3	R	913
101	3	R	1099	3	R	961
102	3	R	1314	3	R	1176
103	3	R	1149	3	R	1011
104	3	R	1382	3	R	1244
105	3	L	1490	3	L	1352
106	3	L	1612	3	L	1474
107	3	L	1763	3	L	1625
108	3	L	1794	3	L	1656
109	3	R	1479	3	R	1341

Table D.5 (continued)

110	3	R	758	3	R	308
111	3	L	572	2	L	2052
112	3	R	1791	3	R	620
113	3	L	1828	3	L	1690
114	4	L	967	4	L	248
115	4	L	1082	4	L	302
116	4	R	1252	4	R	1189
117	4	R	1631	3	R	2074
118	4	R	306	3	R	1996
119	3	L	1887	3	L	1993
120	3	R	1888	3	L	2052
121	4	L	310	4	L	66
122	4	R	372	4	R	66
123	4	L	333	3	R	2019
124	4	L	244	3	L	1934
125	4	L	387	4	L	120
126	4	R	936	4	R	812
127	4	R	1020	4	R	896
128	4	L	1028	4	R	957
129	4	R	1077	4	R	1014
130	3	R	1829	3	R	1649
131	6	L	944	4	L	1450
132	6	R	1516	5	R	1788
133	7	R	625	6	R	625
134	5	R	1805	4	R	1838
135	4	L	1150	4	L	1518
136	5	L	1750	5	L	1915
137	4	R	1680	4	R	1887
138	4	L	1772	4	R	1979
139	5	L	236	5	R	236
140	4	L	1266	4	L	1634
141	4	L	1531	4	L	1899
142	4	L	1680	4	L	2048
143	4	L	1846	5	L	398
144	5	R	332	5	R	568
145	4	R	1576	5	L	324
146	5	L	340	5	L	502
147	5	L	391	5	L	553
148	4	R	1738	4	R	2037
149	4	R	1805	5	R	635
150	4	R	1854	5	R	684

Table D.5 (continued)

151	5	R	439	5	L	660
152	4	L	1884	5	L	698
153	5	L	1137	4	L	2075
154	5	L	1205	5	R	752
155	5	L	1412	5	L	1462
156	5	L	1697	5	L	1747
157	5	L	1833	5	R	1927
158	5	R	1245	4	R	2072
159	6	R	58	5	R	1985
160	5	R	481	5	L	740
161	5	R	891	4	R	1393
162	5	R	617	4	R	1325
163	5	R	549	4	R	1257
164	5	R	720	5	R	855
165	5	R	823	5	R	958
166	5	R	994	5	R	1061
167	5	R	1097	5	R	1164
168	5	R	1200	5	R	1267
169	5	L	459	4	L	370
170	5	L	630	5	L	843
171	5	L	527	4	L	438
172	5	L	733	5	L	946
173	5	L	836	5	L	1049
174	5	L	904	4	L	506
175	5	L	1007	5	L	1152
176	5	L	1110	5	L	1255
177	5	R	1210	5	R	1277
178	6	R	684	6	L	521
179	6	L	1154	6	L	655
180	6	L	1243	6	L	744
181	6	L	1301	6	L	802
182	6	L	1504	6	L	1005
183	6	L	1638	6	L	1139
184	5	L	1886	5	L	2044
185	6	R	392	6	L	334
186	5	R	1885	5	R	2009
187	6	R	468	6	R	701
188	6	L	1020	5	L	1991
189	7	R	817	6	R	893
190	7	R	1173	6	L	1237
191	7	R	1075	6	R	1151

Table D.5 (continued)

192	7	R	1338	6	R	1316
193	7	R	1376	6	R	1354
194	5	R	1360	5	L	1862
195	5	L	1495	5	L	1545
196	5	R	1861	5	R	1844
197	6	R	497	5	R	2038
198	6	R	1005	6	R	1657
199	6	R	702	5	R	2056
200	6	R	1545	6	R	1686
201	6	L	1455	6	L	956
202	6	L	1728	6	L	1327
203	6	L	1821	6	L	1420
204	6	R	1639	6	L	1514
205	6	R	1804	6	L	1679

Table D.6 EMRH Solutions to 205-Task Problem with CT=2266 and CT=2454

	CT = 2266			**CT = 2454**		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	L	1093	1	L	1093
2	1	L	109	1	L	109
3	1	R	690	1	R	690
4	1	L	370	1	L	370
5	1	R	847	1	R	847
6	1	R	1431	1	R	1431
7	1	R	973	1	R	973
8	1	R	1040	1	R	1040
9	1	R	1070	1	R	1070
10	1	R	1176	1	R	1176
11	1	R	1208	1	R	1208
12	1	R	1493	1	R	1493
13	1	L	1147	1	L	1147
14	1	L	1214	1	L	1214
15	1	L	1244	1	L	1244
16	1	L	1350	1	L	1350
17	1	L	1382	1	L	1382
18	1	L	1444	1	L	1444
19	1	L	1500	1	L	1500
20	1	L	67	1	L	67
21	1	R	86	1	R	86
22	1	R	123	1	R	123

Table D.6 (continued)

23	1	R	164	1	R	164
24	1	R	236	1	R	236
25	1	R	322	1	R	322
26	1	L	386	1	L	386
27	1	R	1259	1	R	1259
28	1	R	388	1	R	388
29	1	R	429	1	R	429
30	1	R	919	1	R	919
31	1	R	1310	1	R	1310
32	1	R	1326	1	R	1326
33	1	R	1341	1	R	1341
34	1	L	401	1	L	401
35	1	L	1585	1	L	1585
36	1	L	1644	1	L	1644
37	1	L	2196	1	L	2400
38	1	L	2209	1	L	2413
39	1	L	2265	2	L	185
40	1	R	1896	1	R	1896
41	2	R	859	2	R	670
42	1	R	1976	1	R	1976
43	1	L	2246	1	L	2450
44	2	L	454	2	L	269
45	2	L	472	2	L	287
46	2	L	1017	2	L	832
47	2	L	1276	2	L	1091
48	2	L	1005	2	L	820
49	2	L	1030	2	L	845
50	2	L	1346	2	L	1161
51	2	L	1247	2	L	1062
52	2	L	1418	2	L	1233
53	2	L	1503	2	L	1318
54	1	R	2256	1	R	2445
55	2	R	286	2	R	189
56	2	R	415	2	R	226
57	2	R	428	2	R	239
58	2	R	894	2	R	705
59	2	R	645	2	R	456
60	2	R	966	2	R	777
61	2	R	1051	2	R	862
62	1	R	2001	1	R	2001
63	1	R	2038	1	R	2038

Table D.6 (continued)

64	1	R	2075	1	R	2075
65	1	R	2178	1	R	2178
66	2	R	140	1	R	2318
67	2	R	189	1	R	2367
68	1	R	2213	1	R	2402
69	1	L	1695	1	L	1695
70	1	L	1783	1	L	1783
71	1	L	1836	1	L	1836
72	1	R	1788	1	R	1788
73	1	L	2173	1	L	2173
74	2	L	107	1	L	2280
75	2	R	1702	2	R	1513
76	2	L	204	1	L	2377
77	2	L	370	2	L	166
78	2	L	564	2	L	379
79	2	R	378	2	R	92
80	2	L	670	2	L	485
81	2	L	719	2	L	534
82	2	L	1595	2	L	1410
83	2	L	1966	2	L	1781
84	2	L	806	2	L	621
85	2	L	968	2	L	783
86	2	R	1147	2	R	958
87	2	R	1781	2	R	1592
88	2	R	1243	2	R	1054
89	2	R	1823	2	R	1634
90	2	R	1331	2	R	1142
91	2	R	1913	2	R	1724
92	3	R	97	2	R	2129
93	4	R	270	3	R	1139
94	3	L	1295	3	L	923
95	3	R	210	2	R	2242
96	3	R	548	3	R	338
97	3	L	131	2	L	2163
98	3	R	162	2	R	2194
99	3	L	181	2	L	2213
100	3	L	293	2	L	2325
101	3	L	573	2	L	2373
102	3	L	812	3	L	471
103	3	R	598	2	L	2423
104	3	R	817	3	R	557

Table D.6 (continued)

105	3	L	525	3	L	232
106	3	L	695	3	L	354
107	3	R	749	3	R	489
108	3	L	843	3	L	954
109	3	R	914	2	R	2422
110	3	R	1222	2	R	2032
111	2	L	2082	2	L	1897
112	2	R	2225	3	R	869
113	3	R	1256	3	L	988
114	3	L	1818	3	L	1610
115	3	L	1872	3	L	1664
116	4	R	445	3	R	2132
117	3	L	1927	3	L	1719
118	3	L	1601	3	R	1445
119	3	R	1559	3	L	1291
120	3	R	1618	3	L	1350
121	3	L	1667	3	L	1416
122	3	R	1684	3	L	1482
123	3	L	1690	3	R	1468
124	3	R	1500	3	L	1232
125	3	R	1738	3	R	1522
126	3	R	2032	3	R	1816
127	3	R	2116	3	R	1900
128	3	L	2124	3	L	1916
129	3	R	2173	3	R	1957
130	4	R	619	4	R	1769
131	4	L	944	4	L	1691
132	5	R	1301	4	R	2280
133	5	R	1982	5	R	625
134	4	R	1064	5	R	1070
135	4	L	1012	3	L	2052
136	4	L	2263	5	L	255
137	4	R	1437	3	R	2181
138	4	R	1529	3	R	2273
139	4	R	1765	4	L	501
140	4	L	1128	3	L	2168
141	4	L	1393	4	L	265
142	4	L	1542	3	L	2317
143	4	L	1616	4	L	575
144	4	L	1948	4	R	656
145	4	R	1388	4	R	324

Table D.6 (continued)

146	4	L	2052	4	L	679
147	4	L	2103	3	L	2368
148	4	R	1823	3	R	2331
149	4	R	1890	3	R	2398
150	4	R	1939	3	R	2447
151	4	R	2046	4	R	763
152	4	L	2141	3	L	2406
153	4	L	2210	3	L	2433
154	5	R	583	4	L	747
155	5	R	790	4	R	1731
156	5	L	1259	5	L	202
157	5	L	1342	5	L	338
158	4	R	2081	4	R	2315
159	4	R	2139	4	R	2429
160	4	L	2183	4	R	805
161	4	R	581	4	R	1215
162	4	R	513	4	R	941
163	3	R	2241	4	R	873
164	5	R	103	4	R	1044
165	5	R	206	4	R	1147
166	5	R	309	4	R	1318
167	5	R	412	4	R	1421
168	5	R	515	4	R	1524
169	3	L	1995	3	L	1787
170	5	L	103	4	L	1794
171	3	L	2063	3	L	1855
172	5	L	206	4	L	1897
173	5	L	309	4	L	2000
174	3	L	2192	3	L	1984
175	5	L	412	4	L	2103
176	5	L	515	4	L	2206
177	5	L	525	4	L	2216
178	5	L	1605	5	L	935
179	5	L	1739	5	L	1069
180	5	L	1828	5	L	1158
181	5	L	1886	5	L	1408
182	5	L	1935	5	L	1611
183	5	L	2069	5	L	1745
184	5	L	2122	5	L	1798
185	5	L	1057	5	L	672
186	5	R	2006	4	R	2453

Table D.6 (continued)

187	5	R	2082	5	R	1146
188	5	L	1418	5	L	748
189	6	L	192	5	L	1350
190	6	L	444	5	R	1852
191	6	R	561	5	R	1754
192	6	R	726	5	L	1963
193	6	R	764	5	R	1890
194	5	L	723	4	L	2414
195	5	L	608	4	L	2299
196	5	R	1357	4	R	2371
197	5	R	2111	5	R	1175
198	6	R	303	5	R	1496
199	5	R	2129	5	R	1193
200	5	R	2158	5	R	1919
201	6	L	346	5	L	1562
202	6	L	534	5	L	2053
203	5	L	2215	5	L	2146
204	4	R	2233	5	R	2013
205	6	L	699	5	R	2178

Table D.7 EMRH Solutions to 205-Task Problem with CT=2643 and CT=2832

	CT = 2643			**CT = 2832**		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	L	1093	1	L	1093
2	1	L	109	1	L	109
3	1	R	690	1	R	690
4	1	L	370	1	L	370
5	1	R	847	1	R	847
6	1	R	1431	1	R	1431
7	1	R	973	1	R	973
8	1	R	1040	1	R	1040
9	1	R	1070	1	R	1070
10	1	R	1176	1	R	1176
11	1	R	1208	1	R	1208
12	1	R	1493	1	R	1493
13	1	L	1147	1	L	1147
14	1	L	1214	1	L	1214
15	1	L	1244	1	L	1244
16	1	L	1350	1	L	1350
17	1	L	1382	1	L	1382

Table D.7 (continued)

18	1	L	1444	1	L	1444
19	1	L	1500	1	L	1500
20	1	L	67	1	L	67
21	1	R	86	1	R	86
22	1	R	123	1	R	123
23	1	R	164	1	R	164
24	1	R	236	1	R	236
25	1	R	322	1	R	322
26	1	L	386	1	L	386
27	1	R	1259	1	R	1259
28	1	R	388	1	R	388
29	1	R	429	1	R	429
30	1	R	919	1	R	919
31	1	R	1310	1	R	1310
32	1	R	1326	1	R	1326
33	1	R	1341	1	R	1341
34	1	L	401	1	L	401
35	1	L	1585	1	L	1585
36	1	L	1644	1	L	1644
37	1	L	2400	1	L	2400
38	1	L	2579	1	L	2579
39	1	L	2635	1	L	2635
40	1	R	1896	1	R	1896
41	2	R	481	2	R	431
42	1	R	1976	1	R	1976
43	1	L	2616	1	L	2616
44	2	L	84	1	L	2719
45	2	L	102	1	L	2737
46	2	L	647	1	L	2786
47	2	L	906	1	L	2828
48	2	L	635	1	L	2774
49	2	L	660	1	L	2799
50	2	L	976	2	L	783
51	2	L	877	2	L	713
52	2	L	1048	2	L	855
53	2	L	1133	2	L	940
54	1	R	2537	1	R	2537
55	1	R	2634	1	R	2634
56	2	R	37	1	R	2671
57	2	R	50	1	R	2684
58	2	R	516	1	R	2719

Table D.7 (continued)

59	2	R	267	2	R	217
60	2	R	588	2	R	503
61	2	R	673	1	R	2804
62	1	R	2001	1	R	2001
63	1	R	2038	1	R	2038
64	1	R	2075	1	R	2075
65	1	R	2178	1	R	2178
66	1	R	2318	1	R	2318
67	1	R	2367	1	R	2367
68	1	R	2402	1	R	2402
69	1	L	1695	1	L	1695
70	1	L	1783	1	L	1783
71	1	L	1836	1	L	1836
72	1	R	1788	1	R	1788
73	1	L	2173	1	L	2173
74	1	L	2280	1	L	2280
75	2	R	1324	2	R	1154
76	1	L	2377	1	L	2377
77	1	L	2566	1	L	2566
78	2	L	194	2	L	92
79	1	R	2494	1	R	2494
80	2	L	300	2	L	198
81	2	L	349	2	L	247
82	2	L	1225	2	L	1032
83	2	L	1596	2	L	1403
84	2	L	436	2	L	334
85	2	L	598	2	L	496
86	2	R	769	2	R	599
87	2	R	1403	2	R	1233
88	2	R	865	2	R	695
89	2	R	1445	2	R	1275
90	2	R	953	2	R	783
91	2	R	1535	2	R	1365
92	2	R	1940	2	R	1770
93	3	R	733	3	R	936
94	3	L	574	3	L	816
95	2	R	2282	2	R	2112
96	2	L	2343	2	L	2173
97	2	R	1974	2	R	1804
98	2	L	2005	2	L	1835
99	2	R	2024	2	R	1854

Table D.7 (continued)

100	2	R	2136	2	R	1966
101	2	R	2184	2	R	2014
102	2	R	2399	2	R	2229
103	2	R	2234	2	R	2064
104	2	R	2467	2	R	2297
105	2	L	2575	2	L	2405
106	3	L	122	2	L	2527
107	3	R	151	2	L	2678
108	3	L	605	2	L	2709
109	2	R	2564	2	R	2394
110	2	R	1843	2	R	1673
111	2	L	1712	2	L	1519
112	3	R	463	2	R	2706
113	3	L	639	2	L	2743
114	3	L	1272	3	L	1089
115	3	R	1159	3	R	1047
116	3	R	1712	3	R	1222
117	3	L	1327	3	R	1601
118	3	R	1039	3	R	306
119	3	L	942	2	L	2802
120	3	L	1001	2	R	2803
121	3	L	1067	3	L	310
122	3	R	1105	3	R	372
123	3	L	1090	2	L	2825
124	3	L	883	3	L	244
125	3	L	1144	3	L	364
126	3	R	1453	3	R	666
127	3	R	1537	3	L	900
128	3	L	1524	3	L	961
129	3	L	1581	3	R	993
130	2	R	2602	2	R	2744
131	4	R	1780	4	R	1218
132	4	R	2291	4	R	1729
133	5	R	625	4	R	2410
134	5	R	1070	5	R	445
135	3	L	1649	3	L	1157
136	3	L	2638	4	L	944
137	3	R	2085	3	R	1650
138	3	R	2177	3	R	1742
139	3	R	2413	3	L	1923
140	3	L	1765	3	L	1273

Table D.7 (continued)

141	3	L	2030	3	L	1538
142	3	L	2179	3	L	1687
143	3	L	2253	3	L	1997
144	3	L	2585	3	R	2074
145	3	R	2036	3	R	1546
146	4	L	104	3	L	2101
147	4	L	155	3	L	2152
148	3	R	2471	3	R	2132
149	3	R	2538	3	R	2199
150	3	R	2587	3	R	2248
151	4	R	107	3	L	2259
152	4	L	193	3	L	2297
153	4	L	845	3	L	2811
154	4	L	913	4	R	274
155	4	L	1120	4	L	491
156	4	L	1405	4	L	776
157	4	L	1603	4	L	1027
158	3	R	2622	3	R	2796
159	4	R	2405	4	R	2468
160	4	L	235	3	L	2339
161	4	R	517	3	R	2658
162	4	R	243	3	R	2384
163	4	R	175	3	R	2316
164	4	R	346	3	R	2487
165	4	R	449	3	R	2590
166	4	R	620	3	R	2761
167	4	R	723	4	R	103
168	4	R	826	4	R	206
169	3	L	1395	3	L	2407
170	4	L	338	3	L	2510
171	3	L	1463	3	L	2578
172	4	L	441	3	L	2681
173	4	L	544	3	L	2784
174	4	L	612	4	L	68
175	4	L	715	4	L	171
176	4	L	818	4	L	274
177	4	R	836	4	L	284
178	4	L	2200	4	L	1624
179	4	L	2334	4	L	1758
180	4	L	2423	4	L	1847
181	4	L	2481	4	L	1905

Table D.7 (continued)

182	4	L	2530	4	L	1954
183	5	L	288	4	L	2088
184	4	L	2583	4	L	2141
185	4	L	1937	4	L	1361
186	4	R	2429	4	R	2492
187	4	R	2505	4	R	2568
188	4	L	2013	4	L	1437
189	5	R	1262	5	R	637
190	5	L	1360	5	L	735
191	5	R	1520	5	R	895
192	5	R	1685	5	R	1060
193	5	R	1723	5	R	1098
194	4	L	1520	4	L	891
195	4	L	1203	4	L	574
196	4	R	2347	4	R	1785
197	4	R	2534	4	R	2597
198	5	R	2026	5	R	1401
199	4	R	2552	4	R	2615
200	4	R	2581	4	R	2644
201	5	L	154	4	L	2751
202	5	L	378	5	L	90
203	5	L	471	4	L	2234
204	5	L	565	4	L	2328
205	5	L	730	4	L	2493

Table D.8 EMRH Solutions to 148-Task Problem with CT=204 and CT=255

	CT = 204			**CT = 255**		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	L	16	1	R	53
2	1	R	30	1	R	30
3	1	R	37	1	R	37
4	1	R	84	1	R	100
5	1	L	104	1	R	245
6	1	L	75	1	R	145
7	1	L	143	1	R	184
8	1	R	121	1	R	137
9	1	R	153	1	R	216
10	1	L	172	1	L	202
11	3	L	203	4	L	255
12	5	L	203	7	R	242

Table D.8 (continued)

13	9	L	204	10	L	217
14	1	L	187	2	R	15
15	2	L	53	2	L	154
16	2	R	53	2	R	68
17	2	R	61	2	R	165
18	2	L	117	2	L	212
19	2	R	85	2	R	189
20	2	L	125	2	R	247
21	2	R	134	2	R	254
22	2	L	133	3	L	22
23	2	L	147	3	L	14
24	2	R	147	3	R	13
25	2	R	157	3	R	58
26	2	R	182	3	R	48
27	2	L	158	3	L	58
28	2	L	183	3	L	47
29	2	R	201	3	R	69
30	10	R	166	10	R	250
31	3	R	25	3	R	94
32	1	L	26	1	L	10
33	1	R	167	2	R	82
34	1	L	67	1	L	51
35	2	R	127	2	R	124
36	3	R	72	3	R	141
37	3	R	79	3	R	148
38	3	R	159	3	R	228
39	3	R	166	3	R	235
40	4	R	41	4	R	41
41	11	R	47	10	R	221
42	7	L	68	8	L	39
43	9	L	114	8	L	71
44	10	L	172	8	L	255
45	4	L	80	3	L	242
46	4	L	87	4	L	7
47	4	L	128	4	L	48
48	7	R	193	7	R	255
49	11	L	47	10	L	47
50	3	L	174	1	L	235
51	6	L	34	2	L	188
52	8	L	189	3	L	253
53	11	L	165	8	L	189

Table D.8 (continued)

54	4	L	153	4	L	73
55	4	R	187	4	R	165
56	1	R	195	1	L	79
57	3	L	186	1	L	247
58	7	L	52	6	L	203
59	1	L	201	1	L	133
60	1	R	201	1	R	251
61	1	R	198	1	R	248
62	2	R	190	2	R	205
63	3	R	182	2	R	239
64	4	R	74	2	R	157
65	4	R	82	2	R	197
66	4	R	100	2	R	223
67	4	R	110	3	R	23
68	4	R	124	3	L	72
69	6	R	90	4	R	69
70	3	R	193	8	R	194
71	11	R	165	9	R	118
72	5	R	184	4	R	158
73	2	L	93	1	L	119
74	3	L	40	1	L	173
75	3	L	141	2	L	101
76	4	R	180	4	L	231
77	5	R	28	5	R	28
78	5	R	36	5	R	36
79	7	R	111	4	L	226
80	7	R	118	4	L	238
81	7	R	144	5	L	199
82	7	R	154	5	L	209
83	11	R	186	11	L	43
84	7	R	180	5	L	235
85	2	L	203	4	R	255
86	7	L	204	6	L	255
87	12	R	47	10	L	94
88	12	L	23	9	L	255
89	8	R	203	8	R	255
90	4	L	172	4	L	92
91	7	L	183	5	R	151
92	6	R	125	4	R	200
93	10	L	198	9	L	232
94	12	L	69	10	L	140

Table D.8 (continued)

95	5	R	56	3	R	255
96	9	R	110	7	R	179
97	10	R	185	10	L	236
98	5	R	90	4	R	103
99	4	R	175	3	L	123
100	5	R	129	3	L	162
101	5	R	159	4	R	133
102	9	R	136	7	L	255
103	6	R	189	5	R	253
104	12	R	92	10	L	185
105	9	L	172	5	R	209
106	8	R	28	6	R	28
107	8	L	178	6	R	36
108	9	R	43	6	R	79
109	9	R	176	6	R	189
110	11	L	199	9	R	230
111	4	L	195	4	L	115
112	5	L	162	5	L	162
113	5	L	173	5	L	173
114	6	R	144	6	R	130
115	6	R	203	6	R	237
116	6	R	31	5	R	240
117	6	R	176	6	R	111
118	9	R	202	7	R	205
119	12	L	124	10	R	119
120	6	R	62	6	L	234
121	9	L	32	7	L	202
122	10	L	26	7	R	231
123	6	L	204	6	R	149
124	8	R	190	6	R	251
125	10	R	204	10	L	255
126	12	R	140	8	R	242
127	12	L	179	10	R	174
128	5	L	181	5	L	243
129	5	L	192	5	L	254
130	9	L	59	7	L	229
131	13	L	18	11	L	61
132	9	R	79	2	L	248
133	9	L	82	8	L	23
134	5	R	204	4	R	220
135	10	L	72	8	R	183

Table D.8 (continued)

136	12	R	204	9	L	142
137	12	L	201	11	L	22
138	4	R	202	4	R	235
139	10	L	106	6	R	223
140	13	R	22	9	R	252
141	6	L	185	6	L	151
142	8	R	176	7	R	148
143	13	L	82	9	L	206
144	8	L	170	7	L	170
145	10	R	137	8	R	137
146	13	R	86	10	R	64
147	13	L	160	9	L	78
148	13	R	164	9	R	196

Table D.9 EMRH Solutions to 148-Task Problem with CT=306 and CT=357

	CT = 306			**CT = 357**		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	R	16	1	R	263
2	1	R	168	1	R	315
3	1	R	175	1	R	322
4	1	R	222	2	R	47
5	1	R	251	1	R	351
6	1	R	259	1	L	335
7	1	R	298	2	R	114
8	1	L	306	2	R	151
9	2	R	60	2	R	183
10	2	R	89	2	R	212
11	1	L	191	1	L	283
12	1	L	202	1	L	294
13	7	R	306	7	R	65
14	2	R	104	2	R	227
15	2	L	204	3	L	53
16	2	R	157	2	R	280
17	2	L	212	3	R	220
18	2	L	236	3	L	311
19	2	R	249	3	R	244
20	2	R	257	3	L	319
21	2	R	264	3	R	326
22	2	L	302	3	L	327
23	3	L	14	3	L	341

Table D.9 (continued)

24	2	R	277	3	R	339
25	3	R	259	3	R	351
26	3	R	284	4	R	25
27	3	L	25	4	L	11
28	3	L	50	4	L	36
29	3	R	295	4	L	47
30	6	R	291	1	R	29
31	4	R	162	4	L	72
32	1	L	62	1	L	10
33	1	R	85	1	R	43
34	1	L	103	1	L	51
35	1	R	127	1	R	85
36	4	R	209	5	R	47
37	4	R	216	5	R	54
38	4	R	296	5	R	134
39	4	R	303	5	R	141
40	5	R	41	5	R	182
41	7	R	93	5	R	229
42	1	L	119	1	L	67
43	1	L	151	1	L	99
44	6	L	100	1	L	165
45	5	L	80	5	L	195
46	5	L	87	5	L	202
47	5	L	128	5	L	243
48	5	L	285	7	R	78
49	6	L	147	5	L	290
50	1	L	235	1	L	327
51	1	L	269	2	L	34
52	1	L	162	1	L	176
53	6	L	265	2	L	152
54	5	L	153	5	L	315
55	5	R	164	5	R	357
56	2	R	185	2	R	308
57	1	L	174	1	L	188
58	1	L	52	1	L	240
59	1	R	63	1	R	277
60	1	R	304	1	R	357
61	1	R	301	1	R	354
62	2	R	285	2	L	311
63	3	L	236	2	L	327
64	1	R	49	1	R	118

Table D.9 (continued)

65	1	R	71	1	R	285
66	2	L	254	1	L	353
67	3	L	246	2	L	337
68	3	L	260	3	R	258
69	2	R	28	2	R	75
70	1	R	138	1	R	129
71	7	R	211	1	R	247
72	5	R	189	6	R	25
73	2	R	225	2	R	348
74	2	L	294	3	R	298
75	3	R	249	4	R	341
76	5	L	158	5	L	320
77	5	L	186	5	L	348
78	5	L	194	5	L	356
79	4	L	111	4	L	183
80	4	L	118	4	L	268
81	4	L	144	4	L	294
82	5	L	204	6	R	55
83	5	L	306	7	R	99
84	5	L	230	6	R	81
85	6	L	285	2	L	357
86	2	R	306	3	R	319
87	8	L	229	7	R	146
88	7	L	306	4	L	352
89	6	R	304	7	R	159
90	5	L	249	6	R	100
91	4	L	259	5	L	115
92	4	L	294	4	L	329
93	3	L	306	1	L	266
94	8	L	275	7	R	205
95	3	L	280	5	R	249
96	5	R	72	5	R	280
97	6	L	304	7	R	224
98	5	R	106	5	R	314
99	5	R	157	6	R	151
100	5	R	248	6	R	190
101	5	R	278	6	R	220
102	6	R	26	6	R	246
103	5	R	291	6	R	259
104	9	L	45	7	R	269
105	6	R	84	6	R	317

Table D.9 (continued)

106	6	R	112	6	R	345
107	6	R	120	6	R	353
108	6	R	163	7	L	83
109	6	R	203	7	L	123
110	9	L	79	7	R	303
111	6	R	226	6	L	126
112	7	L	162	6	L	288
113	7	L	173	6	L	299
114	7	L	238	7	L	142
115	7	L	252	7	L	156
116	7	L	283	7	L	187
117	8	R	32	7	L	219
118	8	R	58	7	L	245
119	9	L	134	8	L	55
120	7	R	242	7	L	276
121	7	R	274	7	L	308
122	8	R	84	7	L	334
123	8	R	103	7	R	19
124	8	R	117	7	R	33
125	8	R	278	7	R	322
126	9	L	182	8	L	103
127	9	L	237	8	L	158
128	7	L	181	6	L	307
129	7	L	192	6	L	318
130	7	L	219	6	L	345
131	8	L	18	7	L	18
132	6	R	262	5	R	350
133	5	L	272	6	L	23
134	5	R	209	6	R	45
135	7	R	46	6	L	103
136	9	L	301	8	L	222
137	8	L	40	7	L	40
138	5	R	306	3	L	356
139	6	L	34	6	L	57
140	8	L	297	7	L	356
141	2	L	151	2	L	303
142	3	R	148	3	R	148
143	8	L	104	3	L	287
144	3	L	220	3	L	223
145	4	R	137	4	R	162
146	8	R	181	3	R	212

Table D.9 (continued)

147	8	L	182	4	L	261
148	8	R	259	4	R	240

Table D.10 EMRH Solutions to 148-Task Problem with CT=408 and CT=459

	CT = 408			**CT = 459**		
Task	**Station**	**Side**	**Finish Time**	**Station**	**Side**	**Finish Time**
1	1	R	300	1	L	16
2	1	R	277	1	R	30
3	1	R	284	1	R	37
4	1	R	347	1	R	84
5	1	L	363	1	L	104
6	1	R	355	1	L	75
7	1	L	402	1	L	143
8	1	R	392	1	R	121
9	2	R	32	1	R	153
10	2	R	61	1	L	172
11	4	R	402	2	R	448
12	5	L	408	2	R	459
13	6	R	66	5	L	104
14	2	R	76	1	L	187
15	2	L	374	1	L	240
16	2	R	129	1	R	262
17	2	L	382	1	R	270
18	3	L	24	1	L	332
19	3	R	24	1	R	294
20	3	R	247	1	L	340
21	3	R	254	1	R	348
22	3	L	281	1	L	348
23	3	L	295	1	L	362
24	3	R	267	1	R	361
25	3	R	378	1	R	411
26	4	R	25	1	R	436
27	3	L	306	1	L	373
34	1	L	51	1	L	67
35	1	R	85	1	R	209
36	4	R	136	2	R	72
37	4	R	143	2	R	79
38	4	R	223	2	R	159
39	4	R	230	2	R	166
40	4	R	271	2	R	207

Table D.10 (continued)

41	4	R	318	5	R	150
42	1	L	67	3	L	359
43	1	L	99	4	L	280
44	1	L	165	5	L	170
45	4	L	269	2	L	181
46	4	L	276	2	L	188
47	4	L	317	2	L	229
48	6	R	79	4	R	459
49	4	L	364	5	L	217
50	3	L	121	1	L	450
51	3	L	155	2	L	330
52	1	L	176	2	L	459
53	3	L	273	5	L	335
54	4	L	389	2	L	254
55	5	R	7	2	R	431
56	1	L	294	1	L	268
57	1	L	188	2	L	342
58	1	L	240	2	L	448
59	2	R	143	1	R	341
60	1	R	406	1	L	453
61	1	R	403	1	L	409
62	2	L	390	1	L	417
63	3	R	394	2	R	223
64	1	R	118	1	R	327
65	1	R	400	1	L	406
66	2	L	408	2	R	241
67	3	R	404	2	R	251
68	4	R	332	2	R	265
69	4	R	53	2	R	385
70	1	R	129	1	R	458
71	1	R	247	5	R	268
72	5	R	32	3	R	263
73	1	L	334	1	L	308
74	2	R	183	1	R	401
75	3	R	368	2	L	101
76	4	R	407	2	R	270
77	5	R	103	2	R	298
78	5	R	111	2	R	306
79	4	L	189	3	R	111
80	4	R	339	3	R	118
81	4	R	365	3	R	144

Table D.10 (continued)

82	5	R	121	3	R	154
83	6	R	100	5	R	289
84	5	R	147	3	R	180
85	6	R	186	5	R	309
86	6	L	202	3	R	459
87	6	R	279	5	R	356
88	5	R	406	4	L	459
89	6	R	292	5	L	348
90	4	L	408	2	L	273
91	5	L	115	4	R	263
92	5	R	321	3	R	368
93	1	L	266	5	L	374
94	6	R	338	5	R	402
95	4	R	385	2	L	362
96	6	L	233	4	L	311
97	6	R	357	5	L	393
98	5	R	181	2	L	396
99	3	L	382	2	R	357
100	5	R	220	2	R	424
101	5	R	250	3	R	210
102	6	L	259	4	L	337
103	6	L	272	4	L	350
104	7	L	45	5	L	438
105	6	L	330	4	L	408
106	5	R	278	3	R	238
107	5	R	286	3	R	271
108	6	L	43	3	L	402
109	6	L	370	5	R	40
110	7	L	79	5	R	436
111	5	R	75	2	L	296
112	5	L	277	3	L	162
113	5	L	288	3	L	173
114	6	L	62	3	R	387
115	6	L	384	4	L	422
116	5	R	352	3	R	302
117	6	L	94	3	R	419
118	6	R	126	4	R	446
119	7	L	134	6	R	55
120	5	R	383	3	R	333
121	6	L	126	3	L	434
122	6	R	152	5	L	26

Table D.10 (continued)

123	6	L	145	3	R	438
124	6	R	166	4	L	436
125	6	L	403	5	L	457
126	7	L	182	6	L	48
127	7	L	237	6	L	103
128	5	L	296	3	L	332
129	5	L	307	3	L	343
130	5	L	334	4	L	197
131	5	L	352	6	L	121
132	6	L	181	4	L	233
133	5	L	375	3	L	457
134	5	R	52	4	R	283
135	6	R	232	5	L	72
136	7	L	301	6	R	119
137	5	L	397	6	L	143
138	3	L	397	4	L	248
139	6	R	34	5	R	74
140	6	R	379	5	R	458
141	2	L	151	3	L	324
142	2	R	331	4	R	148
143	3	L	88	6	L	207
144	2	L	321	4	L	170
145	3	R	161	4	R	420
146	2	R	395	6	R	183
147	4	L	78	6	L	285
148	3	R	239	6	R	261

Table D.11 EMRH Solution to 148-Task Problem with CT=510

	CT = 510		
Task	**Station**	**Side**	**Finish Time**
1	1	R	53
2	1	R	30
3	1	R	37
4	1	R	100
5	1	R	174
6	1	R	108
7	1	R	213
8	1	R	145
9	1	R	245
10	1	R	274
11	2	L	321
12	3	R	377
13	3	L	413
14	1	R	289
15	1	L	489
16	1	R	342
17	1	L	497
18	2	L	24
19	2	R	24
20	2	R	32
21	2	R	39
22	2	L	84
23	2	L	98
24	2	R	52
26	2	R	499
27	2	L	109
28	2	L	134
29	2	R	510
30	4	R	471
31	3	L	25
32	1	L	10
33	1	R	356
34	1	L	51
35	1	R	398
36	3	R	195
37	3	R	202
38	3	R	282
39	3	R	289

Table D.11 (continued)

40	3	R	330
41	5	R	47
42	2	L	337
43	3	L	57
44	3	L	215
45	3	L	295
46	3	L	302
47	3	L	343
48	3	L	381
49	3	L	460
50	1	L	239
51	1	L	273
52	1	L	508
53	5	L	118
54	3	L	368
55	3	R	510
56	1	R	426
57	1	L	285
58	2	L	76
59	1	L	65
60	2	L	505
61	1	R	510
62	2	R	60
63	2	R	76
64	1	R	499
65	1	R	507
66	2	R	94
67	2	R	104
68	2	R	118
69	2	R	146
70	3	R	388
71	5	R	165
72	4	R	148
73	1	R	466
74	1	L	105
75	1	L	206
76	3	R	503
77	4	R	51
78	4	R	59
79	2	R	257
80	2	R	264

Table D.11 (continued)

74	1	L	105
75	1	L	206
76	3	R	503
77	4	R	51
78	4	R	59
79	2	R	257
80	2	R	264
81	2	R	290
82	4	R	69
83	4	R	492
84	4	R	95
85	3	L	480
86	3	R	409
87	5	R	212
88	4	L	509
89	4	R	505
90	3	R	498
91	2	L	452
92	2	L	487
93	3	L	506
94	5	R	258
95	2	R	361
96	3	R	440
97	5	R	277
98	2	R	395
99	2	R	341
100	2	R	434
101	2	R	464
102	3	R	466
103	3	R	479
104	5	R	322
105	3	L	115
106	4	R	123
107	4	R	156
108	4	R	199
109	4	L	446
110	5	R	356
111	4	R	23
112	4	L	185
113	4	L	196
114	4	L	296

Table D.11 (continued)

115	4	L	460
116	4	L	235
117	4	L	328
118	4	L	486
119	5	R	411
120	4	L	266
121	4	L	360
122	4	R	428
123	4	L	379
124	4	R	442
125	5	R	430
126	5	R	478
127	5	L	173
128	4	L	204
129	4	L	277
130	4	L	406
131	5	L	191
132	3	R	366
133	4	L	23
134	4	R	219
135	4	R	402
136	5	L	255
137	5	L	277
138	2	L	502
139	3	L	149
140	5	L	299
141	1	L	436
142	3	R	148
143	5	L	363
144	2	L	304
145	4	R	356
146	6	R	64
147	5	L	441
148	6	R	142

APPENDIX E

AMPL CODES OF MATHEMATICAL MODELS

AMPL Code of MM/Int-I

```
param CT;

set T; set L; set R; set E;

param t{T}; param MaxN; param Dir{L union R};

param PreMatrix{T,T};

set P{i in T} = {p in T: PreMatrix[i,p]=1};

param HMatrix{i in T, h in T};

set HR = {i in I, h in I: HMatrix[i,h]=1};

var N >= 0; var Sta{i in T} integer; var FT{i in T}; var z{(i,h) in HR} binary; var AD{i in T}
binary;


minimize length: N;

subject to C11{i in T}:

Sta[i] <= N;

subject to C12{i in T}:

Sta[i] >= 1;

subject to C2{i in I, p in P[i]}:
```

```
Sta[p] <= Sta[i];

subject to C3{i in L union R}:

AD[i] = Dir[i];

subject to C41{i in T}:

FT[i] <= Sta[i] * CT;

subject to C42{i in T}:

FT[i] >= (Sta[i] -1) * CT + t[i];

subject to C5{i in I, p in P[i]}:

FT[i] >= FT[p] + t[i];

subject to C61{(i,h) in HR}:

2*MaxN*CT*(1 - z[i,h]) + MaxN*CT*(AD[h] - AD[i]) + FT[h] >= FT[i] + t[h];

subject to C62{(i,h) in HR}:

2*MaxN*CT *z[i,h] + MaxN*CT*(AD[i] - AD[h]) + FT[i] >= FT[h] + t[i];
```

AMPL Code of MM/Int-II

```
param N; set T; set L; set R; set E;

param t{T}; param MaxCT; param Dir{L union R};

param PreMatrix{T,T};

set P{i in I} = {p in T: PreMatrix[i,p]=1};

param HMatrix{i in T, h in T};

set HR = {i in T, h in T: HMatrix[i,h]=1};
```

```
var CT >=0; var Sta{i in T} integer; var FT{i in T}; var z{(i,h) in HR} binary; var AD{i in T};


minimize cycle: CT;

subject to C11{i in T}:

Sta[i] <= N;

subject to C12{i in T}:

Sta[i] >= 1;

subject to C2{i in T, p in P[i]}:

Sta[p] <= Sta[i];

subject to C3{i in L union R}:

AD[i] = Dir[i];

subject to C41{i in T}:

FT[i] <= CT;

subject to C42{i in T}:

FT[i] >= t[i];

subject to C5{i in T, p in P[i]}:

MaxCT*(Sta[i] - Sta[p]) + FT[i] >= FT[p] + t[i];

subject to C61{(i,h) in HR}:

2*N*MaxCT*(1-z[i,h]) + MaxCT*(Sta[h] - Sta[i] + N*(AD[h] - AD[i])) + FT[h] >= FT[i] +
t[h];

subject to C62{(i,h) in HR}:

2*N*MaxCT*z[i,h] + MaxCT*(Sta[i] - Sta[h]+N*(AD[i] - AD[h])) + FT[i] >= FT[h] + t[i];
```