MULTI CRITERIA ASSEMBLY LINE BALANCING PROBLEM
WITH EQUIPMENT DECISIONS

A THESIS SUBMITTED TO

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

MIDDLE EAST TECHNICAL UNIVERSITY

BY

NİLÜFER PEKİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

INDUSTRIAL ENGINEERING

JANUARY 2006

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Çağlar GÜVEN
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master Science.

Prof. Dr. Meral Azizoğlu
Supervisor

Examining Committee Members:

Prof. Dr. Ömer KIRCA (METU, IE)

Prof. Dr. Meral AZİZOĞLU (METU, IE)

Assoc. Prof. Dr. Yasemin SERİN (METU, IE)

Asst. Prof. Dr. Seçil SAVAŞANERİL (METU, IE)

Prof. Dr. Berna DENGİZ (Başkent U., IE)

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 have fully cited and referenced all material and results that are not original to this work.

Name, Last name **:** Nilüfer, PEKİN

Signature **:**

ABSTRACT

MULTI CRITERIA ASSEMBLY LINE BALANCING PROBLEM WITH EQUIPMENT DECISIONS

Pekin, Nilüfer
M.S., Department of Industrial Engineering
Supervisor: Prof. Dr. Meral Azizoğlu

JANUARY 2006, 76 pages

In this thesis, we develop an exact algorithm for an assembly line balancing problem with equipment selection decisions. Two objectives are considered: minimizing the total equipment costs and the number of workstations. Our aim is to choose the type of the equipment(s) in every workstation and determine the assignment of the tasks to each workstation and equipment type. We aim to propose a set of efficient solutions for each problem and leave the choice of the best solution to the decision maker's preferences. A branch and bound algorithm is developed whose efficiency is increased with some dominance rules and powerful lower bounds. Moreover, modified ranked positional weight heuristic method is used as initial upper bound. The effectiveness of the proposed procedure is demonstrated by computational analysis in which the effects of changing certain parameter values are investigated. We find that our algorithm is capable of solving the problem instances with up to 25 tasks and 5 equipments.

Keywords: Assembly Line Balancing, Equipment Decisions, Branch and bound Algorithm.

ÖZ

EKİPMAN KARARLARI İLE ÇOK KRİTERLİ MONTAJ HATTI DENGELEME PROBLEMİ

Pekin, Nilüfer
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Meral Azizoğlu

OCAK 2006, 76 sayfa

Bu tezde, ekipman kararları ile montaj hattı dengeleme problemleri için algoritma geliştirildi. İki amaç dikkate alındı: toplam ekipman maliyetini ve istasyon sayısını minimize etmek. Her istasyon için ekipman çeşit(leri)ni seçmeyi ve her istasyona atanacak işlere ve işlerin ekipman çeşitlerine karar vermek hedeflendi. Her bir problem için bir etkin çözümler kümesi önerildi ve en iyi çözümün seçimi karar vericinin tercihlerine bırakıldı. Verimliliği bazı eleme mekanizmaları ve güçlü alt limitler ile arttırılan bir dal-sınır algoritması geliştirildi. Ayrıca, modifiye edilmiş sezgisel sıralı konumsal ağırlık metodunu başlangıç üst limiti olarak kullanıldı. Önerilen prosedürün etkinliği belirli parametrelerin etkisinin de araştırıldığı sayısal analizler ile gösterildi. Algoritmanın 25 iş ve 5 çeşit ekipmana kadar olan problem örneklerini çözmeye yeterli olduğu görüldü.

Anahtar kelimeler: Montaj Hattı Dengeleme, Ekipman Kararları, Dal-Sınır Algoritması.

To my family

ACKNOWLEDGEMENTS

I would like to express my gratitude to my Supervisor Prof. Dr. Meral Azizoğlu for the valuable and continual guidance and support she has provided throughout the course of this study. I could not have imagined having a better advisor and mentor, and without her patience, knowledge and perceptiveness I would never have finished.

Most importantly, I would like to express my deepest thanks to my parents, Aynur Pekin and İbrahim Pekin. I am forever indebted to them for their understanding, endless patience, love and encouragement.

I am also grateful to Alper and my sister Yasemin who listened my complaints and motivated me during this study.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

Assembly lines are flow-line production systems, where a series of workstations, on which interchangeable parts are added to a product, are linked sequentially according to the technological restrictions.

Assembly line serve to mass production systems, they consist a number of workstations designed to assemble a specific product or family of products. A product is ready after a complete set of tasks is performed. At each workstation, a subset of the tasks is performed. The product is moved from one workstation to other through the line, and is complete when it leaves the last workstation.

In general, the decision problem, so called assembly line balancing problem, is to find how these tasks are assigned to workstations, so that the predetermined goal is achieved. Minimization of the number of workstations and maximization of the production rate are the most common goals studied in the assembly line balancing literature.

The assembly line systems necessitate continuing improvement due to the shorter life cycle of the products, rapid design changes and growing complexity of the products. With the advent of the new technology, the form of the assembly line balancing systems is adapted to these changes through Flexible Assembly Systems. Flexible Assembly Systems include flexible or automatic equipments, which are capable of performing different tasks, such as robots or flexible machines, like Computer Numerically Controlled machines. The Computer Numerically Controlled machines can perform highly versatile operations provided that the required tools are available in their tool magazines. These tools are generally expensive so that their selection and purchasing may be crucial issue for the effective operation of the flexible assembly systems.

In the flexible assembly systems, developing an efficient flow line is very important. In these systems, the task assignment and equipment selection decisions are made simultaneously. The solution alternatives for sequencing the tasks and selecting the equipment increase rapidly, due to the flexibility brought by the equipments.

In the absence of any technological restrictions, so called precedence constraints, among the tasks and equipment alternatives, the assembly line balancing problem reduces to a sequencing problem for which the number of feasible sequences is $n!$, where n is the number of tasks. When the flexible equipments are added, the number of alternatives increases to $n!*r^n$, where r is the number of equipments. The high number of alternatives necessitates use of an efficient evaluation system en route to find satisfactory solution alternative(s).

Most of the assembly line balancing models assume that the equipments of the workstations are fixed and/or the task times associated to different equipments are the same. Moreover, the studies that consider equipment alternatives ignore cost figures.

In assembly systems, a number of different production alternatives to perform the tasks may exist. Different types of machines, tools or equipments can be used to perform the same tasks and some machinery may be available to a subset of tasks. These decisions have to be considered in assembly systems, since the construction of many assembly lines is a long term decision which requires large investments.

The aim of the many equipment decision problems is the assignment of tasks and equipments to the workstations simultaneously so as to minimize the number of workstations and the system cost including the equipment cost. In the literature, the equipment selection in assembly line balancing problems is frequently referred to as assembly line design problem (ALDP).

In this thesis, we consider single model, single line deterministic assembly line design problem, with equipment selection and task assignment decisions. Not only the assignment of tasks, but also the selection of equipments to the workstations is discussed. There are two main objectives which have to be considered simultaneously: minimization of the total equipment cost and the number of workstations opened. Our aim is to generate a set of efficient, i.e. nondominated, solutions with respect to the total number of workstations and total equipment cost criteria. A branch and bound algorithm, is proposed to find the set of efficient solutions. The best solution is in the efficient set and relative to the decision maker's preferences.

Despite the practical importance of equipment decisions in assembly systems, only few studies in the literature have been considered this issue. We hope our study fills a theoretical gap of the literature.

This thesis includes five chapters that are organized as follows:

In Chapter 2, the terminology used in assembly line balancing is introduced. The literature review on assembly line balancing and equipment decisions are reviewed. Moreover, the mathematical formulation of the problem is introduced.

In Chapter 3, our branch and bound algorithm together with the reduction and bounding mechanisms is described.

In Chapter 4, the computational experiments are conducted to evaluate the performance of the branch and bound algorithm and the results are discussed.

The conclusions, the main results of the study and suggestions for further research directions are presented in Chapter 5.

CHAPTER 2

PROBLEM DEFINITION

In this chapter, we first define the terminology used, overview the assembly line balancing problem, and then give a review of the literature on assembly line balancing problems with equipment selection. Finally we present the mathematical representation of our problem.

2.1 TERMINOLOGY USED FOR ASSEMBLY LINES

Manufacturing a product on assembly lines requires dividing the total work into a set of elementary operations. A task is the smallest, indivisible work element of the total work content. Task time or processing time is the necessary time to perform a task by any specific equipment. The same or different equipments might be required to produce the tasks.

The area within a workplace equipped with special operators and/or machines for accomplishing tasks is called workstation.

Cycle time is the time between the completion times of two consecutive units. Since the tasks are the smallest work elements, in a simple assembly line balancing problem the cycle time cannot be smaller than the largest time of a task.

The work content of a station is the sum of the processing times of the tasks assigned to a workstation.

The tasks are produced in an order due to the technological restrictions that are called the precedence relations or precedence constraints. Processing of a task cannot start before certain tasks are produced. These tasks are known as the predecessors of that task. The successors of a task are the tasks that cannot be performed before the completion of this task. The precedence relations can be represented graphically as illustrated in Figure 2.1.

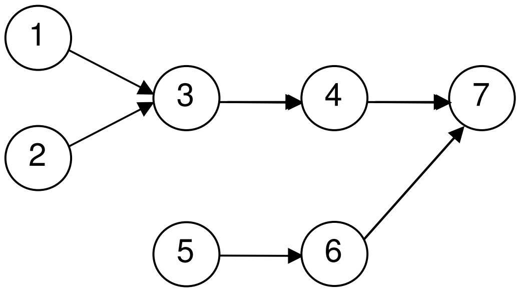


Figure 2.1 An example precedence graph

In the figure, the nodes represent the tasks and an arc between the nodes i and j exists if task i is an immediate predecessor of task j. Accordingly, tasks 1, 2 and 3 are predecessors of task 4 and task 3 is its immediate predecessor. Task 7 is successor of all tasks and an immediate successor of tasks 4 and 6.

Another way of representing the precedence relations is the precedence matrix which is an upper triangular matrix with dimensions labelled by task numbers. If task i is an immediate predecessor of task j then the value of entry (i, j) is 1, otherwise it is 0. The figure below shows the matrix representation of the example, given in Figure 2.1.

Table 2.1 Precedence matrix of the example given in Figure 2.1

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
T_1	-	0	1	0	0	0	0
T_2		-	1	0	0	0	0
T_3			-	1	0	0	0
T_4				-	0	0	1
T_5					-	1	0
T_6						-	1
T_7							-

2.2 AN OVERVIEW OF ASSEMBLY LINE BALANCING

The classical assembly line balancing problem (ALBP) considers the assignment of the tasks to the workstations. Main concern of the assignment is the minimization of the total assembly cost while satisfying the demands and some restrictions like precedence relations among tasks and some system specific constraints.

If a single product is produced on a line, then the problem is called simple assembly line balancing (SALB). In the literature two types of the SALB problems are mainly considered. If the objective is to minimize the total slack time of the line when the cycle time is fixed, the problem is called as SALBP-1 or type-1 ALBP. Minimizing the total slack time is equivalent to minimizing the number of workstations along the line. In the second version of the problem, SALBP-2, the objective is to minimize the cycle time for a given number of workstations. SALBP-2 is also named as type-2 ALBP. Furthermore, some variations in the objectives can be found in the literature such as minimization of the total production cost, minimization of the number of incomplete jobs or maximization of the profit of the system.

Assembly line production systems are utilized to manufacture a large variety of products. As the products have different characteristics, different production systems are necessary to produce them, and therefore, a wide range of assembly line balancing models have been studied.

Since its discovery, assembly line balancing problem has been attracting the interest of many researchers. The main classifications used in the literature are according to the number of the products, the variation of the task times and the operation mode, i.e., paced and unpaced.

There are three kinds of assembly line models according to the products: single model, multi model and mixed model lines. If single model of one product is produced, then the assembly line is called as single model line. In mixed model lines two or more products are manufactured on the same line in an intermixed sequence. The models of the products show small differences so that the same operations are necessary for all products. If various products are produced on the same or several assembly lines, it is known as multi model lines. Different from the mixed model lines the products have significant differences. So, the rearrangement of the line is necessary between switching from one product to another.

Another important classification of the lines is the variation of the task times. The task times are classified as deterministic and stochastic. The automated manufacturing systems or assembly lines which are equipped by flexible machines or robots are assumed to work at a constant speed hence the deterministic task times are well fit. Sometimes the variations of the task times may be significant in affecting the performance of the system; hence the task times are stochastic. When the lines are operated manually, the variations of the task times are expected due to the skills and motivations of the employees. Moreover, due to the learning effects or successive improvements of the production process variations between the task times may occur.

Depending on the operation mode of the workstations, the flow lines may be paced or unpaced lines. If the assembly line in which the time spent in each workstation is fixed and same for all workstations, the system is known as the paced assembly line. In paced assembly lines, if the maximum processing time is larger than the cycle time, then the parts pass to the next workstation although it is incomplete. In an unpaced assembly line, unlike to the paced lines, the time spent in each workstation is different. Due to the fact that all workstations operate at individual speeds, the buffer stocks may be required between the workstations.

In the literature, there are several models and many different solution procedures that have been introduced to solve the assembly line balancing problem. These solution procedures can be classified as exact and heuristic methods. The exact methods are branch and bound algorithms, integer programming solutions and dynamic programming procedures. On the other hand, a large variety of heuristic methods, like priority based procedures, incomplete enumeration procedures and search methods are proposed.

The most recent reported survey papers on the assembly line balancing problem are due to Baybars (1986), Ghosh and Gagnon (1989), Scholl and Becker (2003) and Becker and Scholl (2003).

Baybars (1986) defines the simple assembly line balancing problem (SALBP) with some modifications and generalizations over time. A summary of the deterministic models, the exact solution algorithms and integer programming formulations are discussed comprehensively.

Ghosh and Gagnon (1989) present a literature review and analysis of the assembly line balancing and scheduling of assembly systems. Quantitative developments and qualitative issues are discussed at the strategic and tactical levels. They classify the assembly line balancing problems in four classes: single model deterministic, single model stochastic, multi/mixed model deterministic and multi/mixed model stochastic. The literature review of simple and general cases of each of these problems is discussed. The methodologies as well as the objective criteria are also presented. Moreover, eight important factors that effect the design and balancing of the assembly systems are stated. These are output focus, line type, process and equipment considerations, facility considerations, workstation considerations, task-related considerations, worker related and schedule related considerations. The factors organized in hierarchical and factor/design taxonomy are defined to access the progress in assembly line balancing.

Scholl and Becker (2003) discuss a comprehensive survey of simple assembly line balancing problems. Exact and heuristic procedures for all the problem types are given in detail with an emphasis on the significant algorithmic developments.

The review of generalized assembly line balancing problems (GALBP) is discussed by Becker and Scholl (2003). The generalized problem with additional characteristics such as cost functions, equipment selection, paralleling and U-shaped line layout and mixed model production are reviewed. In addition, the recent developments on the sophisticated solution procedures of the models are presented.

2.3 LITERATURE REVIEW OF EQUIPMENT DECISIONS IN ASSEMBLY LINE BALANCING

In the literature, several versions of the assembly line balancing problem are studied some of which consider the equipment alternatives. However, there are only few studies that address the task and equipment assignments together.

Graves and Whitney (1979) develop an optimization method for equipment selection problem. The aim is to select the equipments and assign the tasks in order to minimize the system cost. The system cost includes the annual fixed costs of workstations and operating costs. It is assumed that there are a finite number of workstations which are not identical. A mixed-integer linear program is formulated for a single product that has a fixed sequence of tasks. A branch and bound algorithm with a subgradient optimization procedure is proposed to solve the problem.

Graves and Lamar (1983) extend the model of Graves and Whitney (1979) so as to include equipment change times. As the integer program developed is very large, an approximate solution procedure for finding the lower and upper bounds is discussed.

Pinto et al. (1983) present a model that considers the choice of the manufacturing alternatives and the assignment of tasks so as to minimize the total costs which is the sum of the labour cost and the fixed expenses. The model describes a process which may be complemented by one or more process alternatives each of which reduces some task times or even removes certain tasks completely. The combined processing alternative line model is formulated by integer programming. Two different formulations that differ in the degree of flexibility in selecting the cycle time are presented.

Graves and Holmes Redfield (1988) consider the equipment selection model of Graves and Lamar (1983) with some modifications. Their design problem consists of task assignments of one or several products with tool costs and tool change times. The problem is solved by an optimization procedure that assigns tasks to workstations and selects the assembly equipment for each workstation.

Rubinotitz and Bukchin (1993) present a heuristic approach for designing and balancing a robotic assembly line. The objective is to minimize the number of workstations and robots used. Several robot alternatives are available for each task. The balancing problem is simplified by the restriction that single equipment to each workstation is allowed. In addition it is assumed that all the equipments have identical purchasing costs. A branch and bound frontier search method is used as the base of the heuristic algorithm.

Bukchin and Tzur (2000) develop an optimization and heuristic algorithms for the design of flexible assembly lines. The goal is minimizing the total equipment cost by selecting the equipments and assigning tasks to workstations. Several equipment alternatives, which have different costs and effects on the task times of the product, are given for each task. As the majority of the literature on equipment selection, the assignment of one equipment is allowed in each workstation. A branch and bound algorithm is proposed to find the exact solutions. Their heuristic procedure is a version of the branch and bound algorithm, which skips some nodes by user specified parameters.

Rekiek et al. (2002a) present a hybrid assembly line design. Two objectives are considered: minimizing the total cost and integrating design and operation issues. Different from the equipment selection models, operating modes of the equipments are defined such that manual, robotic and automated. The model is solved by branch and cut method and the multicriteria decision aid method PROMTHEE II. Firstly the tasks are assigned to the workstations according to the equal piles strategy, and then all possible resource combinations for each workstation are generated by the branch and cut algorithm. Finally the best possible combination is selected by the PROMTHEE II for a single product.

An equipment selection problem with parallel workstation case is developed by Bukchin and Rubinotitz (2002). Similar to the previous studies, minimizing the number of workstations and the total cost is discussed. The model is presented as a special case of equipment selection problem with the assumption that the task times may exceed the cycle time. A branch and bound optimal algorithm is developed for finding the exact solution.

The most closely study to our study is due to Bukchin and Tzur (2000). Our study differs from the Bukchin and Tzur (2000)'s in the following senses.

In our study,

- More than one equipment can be assigned to a single workstation,
- Two objectives, minimizing total equipment cost and total number of workstations, are considered,

- The set of efficient, i.e., nondominated, solutions are considered relative to the two objectives,
- The choice of the optimum solution from the efficient set depends on the preferences of the decision maker.

2.4 PROBLEM DEFINITION

In this study, deterministic single model line is considered, i.e., all input parameters are given and assumed to be known with certainty. One product is continuously manufactured on a line. Task times, precedence relations of the tasks, cycle time and costs of the equipments all together define the problem data. We suppose that the processing times of the tasks vary with respect to the flexible equipments, which are able to perform many different tasks. We assume there is at least one equipment with which each task can be performed.

For simplicity, we index the equipments with respect to their costs. Accordingly, the first equipment indexed as E_1 is the cheapest and the last equipment E_r is the most expensive one.

2.4.1 MATHEMATICAL FORMULATION

In this section, we present our assumptions, the notation and the mixed integer programming formulation of the problem.

Our assumptions are listed below:

- A single product is assembled on the line.
- The processing times of tasks are deterministic and depend on the equipment selected to perform the task.
- The assembly tasks cannot be split.

- Material handling, loading and unloading times are negligible or included in the task durations.
- The cycle time of the workstations is known and is not subject to change.
- The precedence relations between assembly tasks are known.
- The task process times are independent of the workstations and of the succeeding and/or preceding tasks.
- There is a given set of equipment types, each type has a known specific cost that includes the purchasing and the operational costs.
- The equipments costs are same for all tasks.
- The set up times of performing tasks are negligible or included in the task times.
- A task can be performed at any workstation of the assembly line, provided that the equipment selected for this workstation is capable of performing the task, and that precedence relations are satisfied.
- More than one equipment can be assigned to each workstation on the line.

The notation used in the mathematical formulation of the problem is given below.

Indices:

i = task index
k = equipment index
g = workstation index

The problem is defined by the following parameters:

n = number of tasks
r = number of equipments
C = cycle time
t_{ik} = duration of task i when performed by equipment k
EC_k = cost of equipment k

Decision variables:

$$x_{ikg} = \begin{cases} 1 & \text{if task i is performed in workstation g by equipment k,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{kg} = \begin{cases} 1 & \text{if equipment k is assigned to workstation g,} \\ 0 & \text{otherwise.} \end{cases}$$

ST = number of workstations opened.

The mixed integer programming formulation of the problem is given below:

$$\text{Min } f \left(ST, \sum_{k=1}^{r} \sum_{g=1}^{n} EC_k \, y_{kg} \right) \tag{1}$$

Subject to

$$\sum_{k=1}^{r} \sum_{g=1}^{n} x_{ikg} = 1 \qquad \forall i \tag{2}$$

$$\sum_{k=1}^{r} \sum_{i=1}^{n} t_{ik} x_{ikg} \leq C \qquad \forall g \tag{3}$$

$$\sum_{k=1}^{r} \sum_{g=1}^{n} g \;\, x_{akg} \leq \sum_{k=1}^{r} \sum_{g=1}^{n} g \;\, x_{bkg} \qquad \forall (a,b), \text{ such that a immediately precedes b} \tag{4}$$

$$\sum_{k=1}^{r} \sum_{g=1}^{n} g \;\, x_{ikg} \leq ST \qquad \forall i \tag{5}$$

$$x_{ikg} \leq y_{kg} \qquad \forall i,k,g \tag{6}$$

$$x_{ikg} = 0,1 \qquad \forall i,k,g \tag{7}$$

$$y_{kg} = 0,1 \qquad \forall k,g \tag{8}$$

$$ST \geq 0 \tag{9}$$

The objective function (1) represents a function of the equipment cost and the number of workstation to be minimized.

Constraint set (2) ensures that all the tasks are assigned only once.

Constraint set (3) is the capacity constraint and guarantees that the work content of every workstation is no longer than the prespecified cycle time.

Constraint set (4) ensures the precedence relations between the tasks a and b, such that if task a immediately precedes b, then task a cannot be assigned to later workstation than task b's station.

Constraint set (5) ensures that the assignment of all the tasks necessitates at least ST workstations.

Constraint set (6) represents the relationship between the variables x_{ikg} and y_{kg} by not allowing any task to be performed on a workstation if its equipment is not assigned to the workstation.

Constraint set (7) sets the decision variable x_{ikg} to binary values.

Constraint set (8) defines the choices for y_{kg} , however the set is redundant due to the existence of set (7).

Moreover, the constraint set (5) lower limits the variable ST hence the constraint set (9), i.e., $ST \geq 0$, is also redundant.

A solution ES is said to be efficient with respect to two criteria, number of workstations, ST and total equipment cost, $\sum_{k=1}^{r}\sum_{g=1}^{n} EC_k y_{kg}$ if there exists no solution

ES' with $\sum_{k=1}^{r}\sum_{g=1}^{n} EC_k y_{kg}(ST') \leq \sum_{k=1}^{r}\sum_{g=1}^{n} EC_k y_{kg}(ST)$ and, $ST'(ES') \leq ST(ES)$ strict inequality holding at least once. If solution ES' exists then ES is said to be inefficient, i.e., dominated solution.

There is an optimal solution in the efficient set as long as the objective function is a monotone increasing function of ST and $\sum_{k=1}^{r}\sum_{g=1}^{n} EC_k y_{kg}$.

As long as $f\left(ST, \sum_{k=1}^{r}\sum_{g=1}^{n} EC_k y_{kg}\right)$ is monotone increasing and known, the above program can be used to find an optimal solution. If moreover f is linear function of ST and $\sum_{k=1}^{r}\sum_{g=1}^{n} EC_k y_{kg}$ then the model is mixed integer linear program.

When f is a monotone increasing function but unknown, one has to generate all efficient solutions. The optimal solution for any f , is in the efficient set.

When $f\left(ST, \sum_{k=1}^{r}\sum_{g=1}^{n} EC_k y_{kg}\right) = ST$ and $t_{ik} = t_i \quad \forall k$, the problem reduces to the simple assembly line balancing problem (SALBP). The SALBP is an NP-hard problem so is our problem to minimize the unknown monotone increasing function.

2.4.2 AN EXAMPLE PROBLEM

We illustrate our equipment selection and task assignment problem on a simple example. The example consists of 10 tasks and 3 equipment alternatives that are capable of performing all tasks. We assume the cycle time is 40 time units. Table 2.4 illustrates the times required to produce the tasks and shows the equipment costs.

Table 2.2 The task times and the equipment costs of example I

Equipment k / Task i	1	2	3
1	9	18	12
2	21	5	6
3	12	12	7
4	13	13	8
5	22	24	15
6	24	8	12
7	9	5	13
8	16	17	17
9	21	19	20
10	25	18	18
Equipment cost	50	90	120

The following figure depicts the precedence structure.

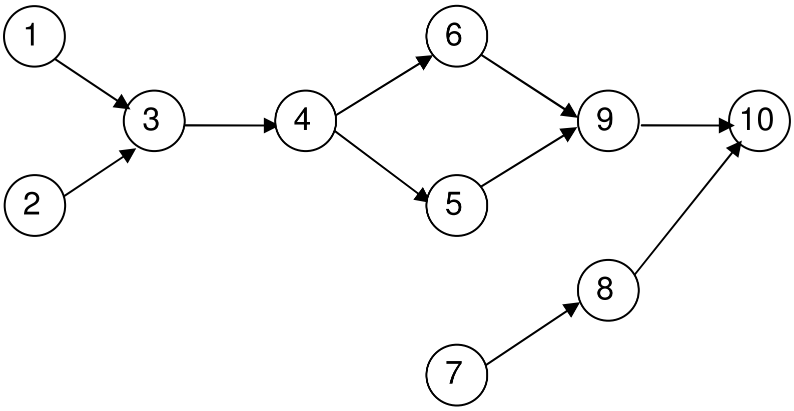


Figure 2.2 The precedence graph of example I

There are two efficient solutions to the problem as depicted by the following configurations:

Solution 1:

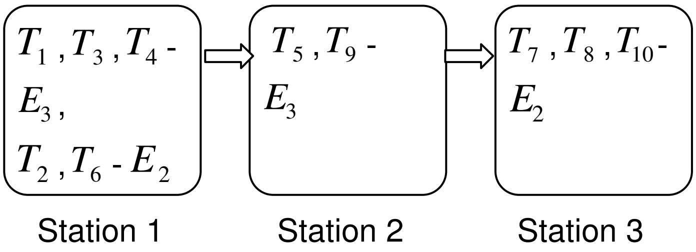


Figure 2.3 The first efficient solution of example I

Total equipment cost = $EC_3 + EC_2 + EC_3 + EC_2 = 120 + 90 + 120 + 90 = 420$

Number of workstations = 3

Solution 2:

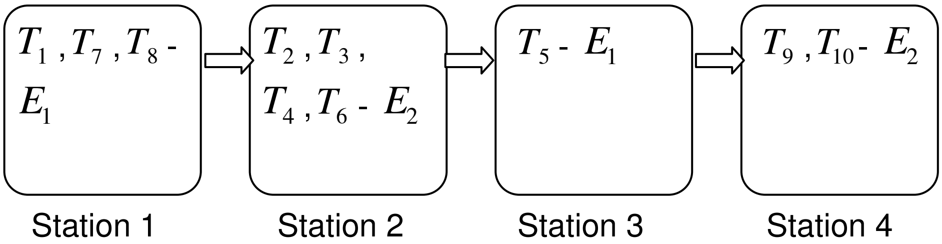


Figure 2.4 The second efficient solution of example I

Total equipment cost = $EC_1 + EC_2 + EC_1 + EC_2 = 50 + 90 + 50 + 90 = 280$

Number of workstations = 4

The trade off between the alternatives can be set by considering the number of workstations and the total equipment cost. The first solution is favoured by a decision maker who penalizes the number of workstations more than the total equipment cost. On the other hand, Solution 2 is favoured by a decision maker who penalizes total equipment cost more than the number of workstations.

CHAPTER 3

BRANCH AND BOUND ALGORITHM

Our problem of generating all efficient solutions is NP-hard as it reduces to the well-known NP-hard problem of minimizing the number of the workstations. This justifies use of implicit enumeration techniques like branch and bound algorithm and dynamic programming procedures.

In this study we propose a branch and bound algorithm to find all the efficient solutions with respect to the number of the workstations and the total equipment cost criteria.

Depth-first search method is used to guide the search in the branch and bound algorithm. According to this strategy, a single branch of the tree is developed until a feasible solution is reached. In each branching point the nodes are generated and the node with the minimum cost is selected for the next branching. The nodes, which are not eliminated, are sorted and stored in a stack in the nondecreasing order of their costs for backtracking.

We first produce as a set of approximate efficient solutions for initial upper bounds. We let $UB(g)$ be the total equipment cost of a feasible solution with g workstations. We update $UB(g)$ whenever a solution with g workstations and smaller total equipment cost is found. We fathom the node having g workstations if the associated equipment cost is greater than $UB(g)$. Whenever the algorithm terminates $UB(g)$ is the minimum total equipment cost overall solutions having g workstations. Our algorithm stops whenever all nodes are searched.

In the solution list, the total equipment cost of the feasible solutions decreases as the number of workstations increases. If such a decrease in the cost value is not observed, then the solution is not recorded as efficient, since the decision maker always prefers the smaller cost with the fewer number of workstations. Table 3.1 shows a sample solution list of a problem. According to the table, four different feasible solutions are available.

Table 3.1 An example solution list

g	$UB(g)$
3	1500
4	1300
5	1200
6	1100

We develop some procedures to improve the efficiency of the branch and bound algorithm. These are reduction mechanisms, lower bounds and initial upper bound procedures. The reduction mechanisms, i.e. the node elimination mechanisms, for reducing the size of the solution tree are discussed in the next section.

3.1 REDUCTION MECHANISMS

We develop some mechanisms in order to increase the efficiency of our branch and bound algorithm. The node elimination mechanisms are presented in three sets: branching scheme properties, problem reduction conditions and node fathoming conditions.

3.1.1 BRANCHING SCHEME PROPERTIES

The branching schemes for simple assembly line balancing problem work as follows: at each level, an assignment of an unscheduled task to the current workstation is considered. If a task cannot fit to the current workstation due to the cycle time constraint, then the resulting solution corresponds to opening a new workstation. The candidate tasks for assignment are the ones whose predecessors are already appeared in the current node, i.e., partial solution.

Our problem has equipment assignment decisions in addition to the task assignment decisions. So we have to consider the assignments in pairs, each pair corresponding to an unassigned task and a particular equipment. Moreover we have to decide to close or not to close the current workstation even the task fits in it. Assume we have two unassigned tasks say T_i and T_j and two equipment alternatives E_k and E_l, the resulting eight decisions are shown in the tree below.

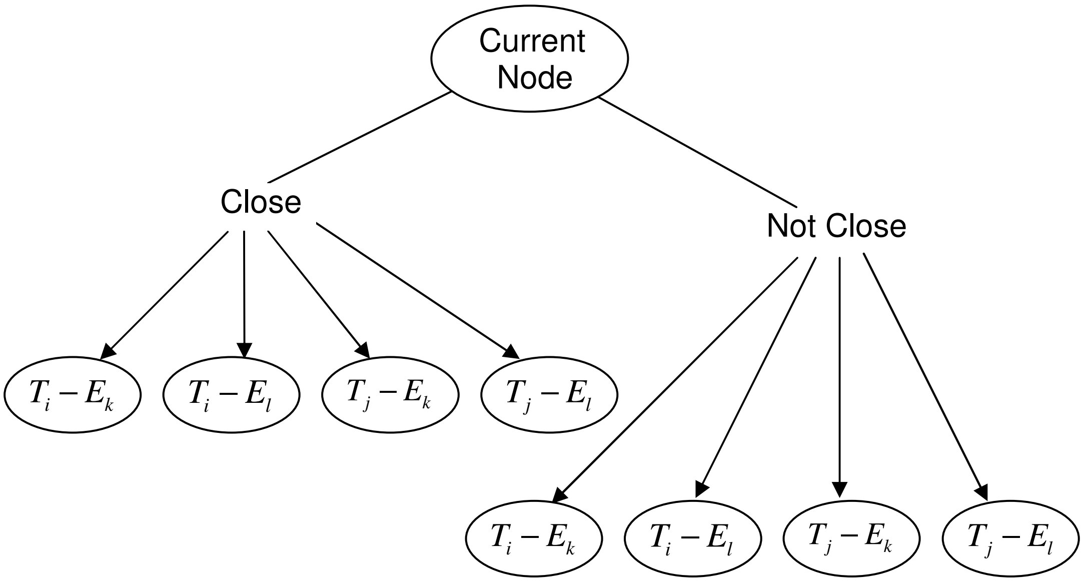


Figure 3.1 An example branching tree

The size of the branch and bound tree is reduced by using the results of branching scheme properties, stated in Property 1 and Property 2.

Property 1:

If there exists any fittable task with no additional equipment requirement, then never branch to a node that represents opening a new workstation.

Proof:

Assume the new workstation g+1 is opened even when there is a fittable task with no extra equipment requirement, say task i. Assume task i is assigned to workstation g+1. Task i can be removed from workstation g+1 and replaced into workstation g without increasing the number of workstations and total equipment cost as it fits to workstation g with no extra equipment. Hence a solution in which task i is replaced into workstation g while keeping the other assignments cannot be worse. □

Property 2:

A node that assigns task i and E_k to the current workstation is fathomed if $t_{ik} \geq \min_{l \in A} \{t_{il}\}$ where A is the set of equipments already assigned to the current workstation.

Proof:

Assume $t_{ik} \geq \min_{l \in A} \{t_{il}\} = t_{is}$ where A is the set of equipments already assigned to the current workstation. A node that assigns T_i together with E_k is dominated by the node that assigns task T_i together with E_s. This due to the fact that $t_{ik} \geq t_{is}$ and equipment s is already in the workstation. Hence assignment of T_i with E_k never produces fewer number of workstations and smaller total cost than the assignment of the combination of T_i and E_s. □

Using the result of property 2, we consider at most two types of nodes for the assignments in the current workstation for T_i .

Node 1: Assignment of T_i with E_k where $t_{ik} = \min_{l \in A} \{t_{il}\}$ to the current workstation.

Node 2: Assignment of T_i with E_k for all k such that $k \in$ A' where A' is the set of equipments that are not already assigned to the current workstation and $t_{ik} < t_{il}$ where $l \in$ A.

Example II

In this section, we present a small example that shows the power of properties 1 and 2 in eliminating the partial solutions. Assume an assembly system with 4 tasks and 3 equipments. The time required of each task by each equipment and the precedence relations of the tasks are given in the Table 3.2 and Figure 3.2 respectively.

Table 3.2 The task times and the equipment costs of example II

Equipment k / Task i	1	2	3
1	7	3	2
2	7	8	8
3	9	4	-
4	5	10	8
Equipment cost	8	10	10

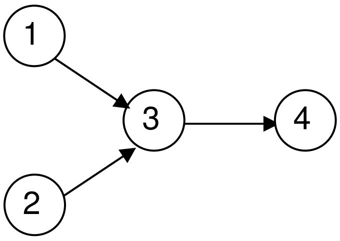


Figure 3.2 The precedence graph of example II

We assume the cycle time is 15 time units. In our branching scheme, the possible task-equipment pairs are generated using properties 1 and 2. Figure 3.3 illustrates a part of the solution tree of example II.

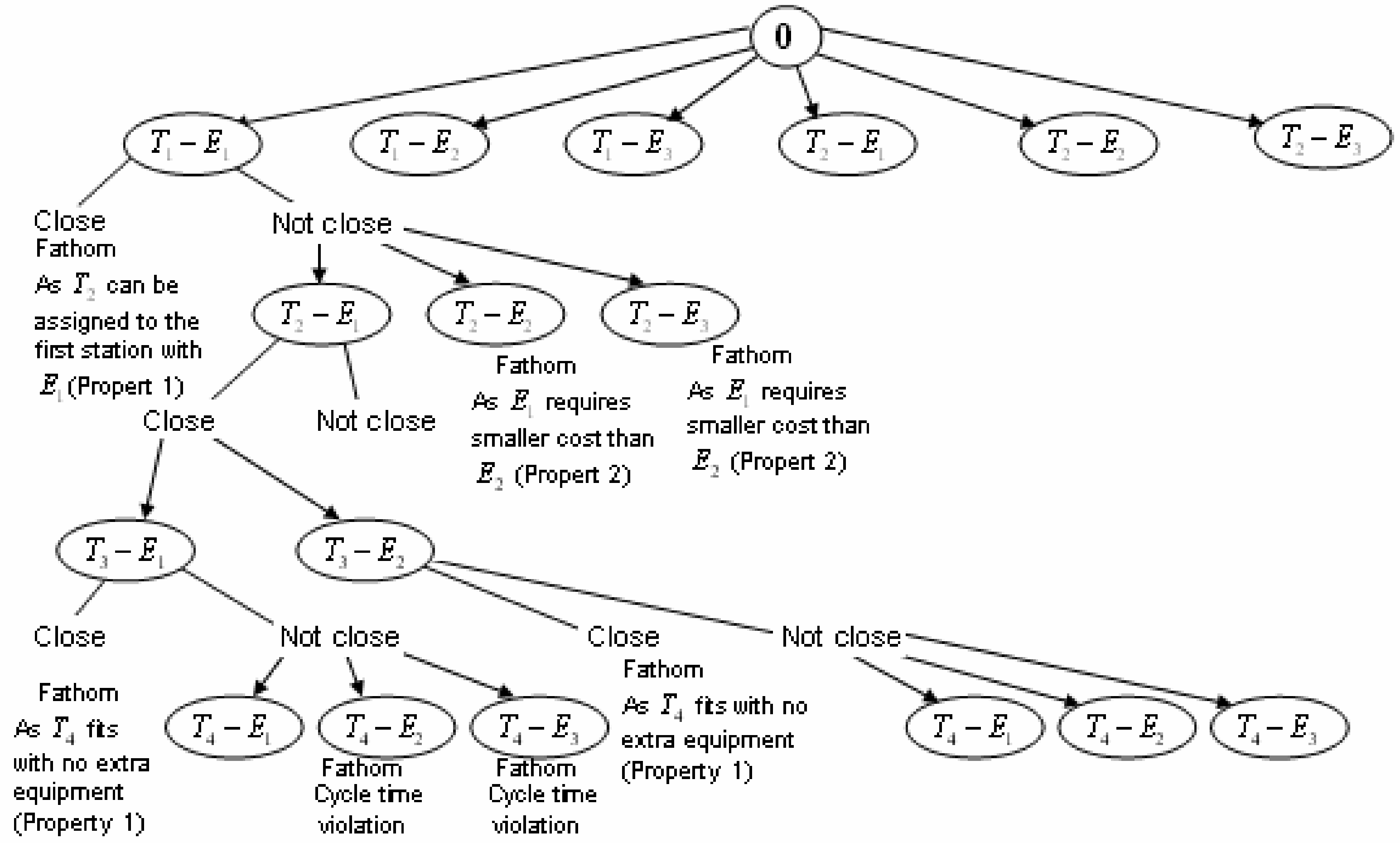


Figure 3.3 Branching scheme of example II

We use the results of Property 1 and Property 2, in generating nodes as follows:

Let A (A') be the set of equipments already (not yet) assigned to the current workstation. Among A, we branch to a single node, for task i. The generated node considers the assignment of the task to equipment E_l such that $t_{il} = \min_{k \in A} \{t_{ik}\}$.

Among A', for task i, we only branch to the nodes that yield lower task times than t_{il} where $l \in$ A. So the generated nodes consider the assignment of task i to equipment k such that $t_{ik} < t_{il}$ for all A' and $l \in$ A.

3.1.2 PROBLEM REDUCTION PROPERTIES

In this section, we present two properties that are used to reduce the size of the problem.

Property 3**:**

If $\min_l \{t_{il}\} + \min_l \{t_{jl}\} > C$ for all tasks j, then task i is assigned to a workstation singly, with equipment k where $EC_k = \min_l \{EC_l \mid t_{il} < C\}$.

Proof:

If $\min_l \{t_{il}\} + \min_l \{t_{jl}\} > C$ then task i cannot be assigned to any workstation with task j. If this holds for all tasks j then task i cannot be assigned to any workstation with any one of the tasks, hence should be assigned to a workstation with no other assignments. An equipment assignment does not violate the cycle time constraint, as we assume $t_{ik} < C$, for all i and k. Among the feasible assignments, only equipment(s) having the smallest cost leads to the optimal cost. □

Property 3 can be used to reduce the size of the problem before solving the problem. Moreover for a partial assignment, one can check the condition for all unassigned tasks and remove the size of the remaining problem.

Property 4**:**

If $EC_k \geq EC_l$ and $t_{ik} \geq t_{il}$ for all tasks i then there exists an optimal schedule in which E_k is not assigned to any workstation.

Proof:

Assume an optimal schedule OS in which E_k is assigned to one of the workstations. Replacing E_k with E_l does not increase the number of workstations as $t_{ik} \geq t_{il}$ for all i. Moreover such an exchange does not increase total cost as $EC_k \geq EC_l$. Hence OS cannot be a unique optimal solution. □

Property 4 can be used to reduce the size of the problem by removing E_k. Moreover we can employ the property for any partial solution as follows:

If $EC_k \geq EC_l$, E_k is not assigned to the current workstation and $t_{ik} \geq t_{jl}$ for all unassigned tasks i then we can remove E_k from all future assignments and bound calculations.

3.1.3 NODE ELIMINATION PROPERTIES

In this section, we introduce some properties that help to reduce the size of the search by eliminating some nodes without being evaluated.

Property 5**:**

If $\sum_{i\in T'}\min_{k\in A}\{t_{ik}\}\le S$, where T' is the set of unassigned task and S is the total idle, i.e. slack, time in the current workstation then all tasks in T' are put to the current workstation, is an optimal solution emanating from the current node.

Proof:

If $\sum_{i\in T'}\min_{k\in A}\{t_{ik}\}\le S$ then all tasks can be put to the current workstation with no additional workstation opening and equipment costs. Hence this assignment is optimal for the remaining tasks. □

If the conditions of the above property hold then we put all the tasks to the current workstation, update the current best known solution, if necessary, and backtrack.

Property 6**:**

If $\sum_{i\in T'}\min_{k}\{t_{ik}\}> S$ and $\sum_{i\in T'}t_{i1}\le C$, where E_1 is the cheapest equipment, then there is an optimal solution in which all tasks are assigned to the next workstation with equipment E_1 , is an optimal solution emanating from the current node.

Proof:

Note that even the smallest task times are incurred; all tasks cannot fit to the current workstation. Hence a lower bound on the number of remaining workstations is 1 and a lower bound on the total equipment cost is EC_1 . As $\sum_{i\in T'}t_{i1}\le C$, it is possible to complete all tasks are realized in the next workstation with equipment E_1 , i.e. lower bound. □

If the conditions of the above property hold we increase the number of workstations by one and the total equipment cost by EC_1 units, update the current best known solution, if necessary, and backtrack.

Properties 5 and 6 should be checked in each iteration: when a new task is scheduled.

Property 7**:**

If $C-\sum_{i\in WT}\min_{k\in A}\{t_{ik}\}>S$, where WT is the set of tasks assigned to the current workstation then the current solution cannot yield to a unique optimal solution.

Proof:

If the condition of the property holds, then at least one task is not already assigned to its minimum time equipment. Assigning each task to its minimum time equipment increases the slack time of the workstation and may leave one of the equipments idle. Higher slack time and more vacant equipments may decrease but never increases the number of workstations and total equipment cost respectively. Hence the current solution, in which at least one task is not assigned to its minimum time equipment cannot lead to a unique optimal solution. □

We use property 7 whenever closing a workstation if the conditions of the property hold then we fathom the node.

Property 8**:**

If an assigned equipment E_k can be replaced by E_l such that $EC_k \geq EC_l$ without violating the cycle time constraint then the current assignment cannot lead to a unique optimal solution.

Proof:

As E_l can be exchanged by E_k without violating the cycle time constraint then the resulting solution cannot have higher number of workstations. Moreover the total equipment cost is never larger as $EC_k \geq EC_l$. Hence the current assignment cannot lead to a unique optimal solution. □

We use the result of Property 8 whenever closing a workstation. If E_k is assigned to the current workstation, but not E_l then we fathom the node.

Moreover if we can replace the equipment of any task assigned to the current workstation with any cheaper assigned equipment then the current assignment cannot yield to a unique optimal solution, thus can be fathomed.

Property 9**:**

If an assignment equipment E_k can be replaced by E_l such that $EC_k = EC_l$ without violating the cycle time constraint and decreasing the slack time of the workstation then the current assignment cannot lead to a unique optimal solution.

Proof:

As replacement by E_l results with increased slack time, it may decrease but never increases the number of workstations. The total equipment cost does not change after replacement as $EC_k = EC_l$. Hence the current solution cannot lead to a unique optimal solution. □

We use the result of Property 9 whenever closing a workstation. If E_k is assigned to the current workstation and replacing E_k with E_l leaves no smaller slack time then we fathom the node.

Property 10**:**

Assume E_k and E_l are two equipments assigned to the current workstation. If any task i is assigned to E_k, but can be replaced by E_l, without violating cycle time constraint and if either $EC_k \geq EC_l$ or $t_{ik} \geq t_{il}$ then the current assignment cannot lead to a unique optimal solution.

Proof:

Note that if $EC_k \geq EC_l$ and E_k is assigned to task i then exchanging the equipment of task i to E_l may decrease, but never increases the total equipment cost. The number of workstations does not change, as the solution after the exchange is feasible as well. Moreover if $t_{ik} \geq t_{il}$, then exchanging the equipment of task i to E_l may increase the total slack time, which in turn may decrease the number of workstations. The equipment cost also may decrease if such an exchange leaves E_l unassigned. Hence the current assignment cannot yield to a unique optimal solution.

□

We use the result of Property 10 whenever an equipment is assigned to the current workstation.

3.2 INITIAL UPPER BOUND PROCEDURE

We find an initial approximate set of efficient solutions by modifying the ranked positional weight heuristic method designed for simple assembly line balancing problem.

The ranked positional weight heuristic orders the tasks in descending order of their positional weights. The positional weight of a task is the sum of the task time of the task and task times of all its successors. In each iteration, a task with highest priority is assigned to the current workstation if it fits, otherwise the current workstation is closed and a new one is opened. The procedure terminates whenever all tasks are assigned.

We implement the ranked positional weight r times, each time using the task times associated to a particular equipment.

Similar to the branch and bound algorithm the procedure of the heuristic method is modified in order to improve the accuracy of the method.

Whenever closing a workstation for a problem of equipment E_l we modify the equipment assignment as follows:

If there exists E_k such that $EC_k < EC_l$ and $\sum_{i \in WT} t_{ik} \leq C$ where WT is the set of tasks assigned to current workstation, we replace E_k with E_l. Note that such a replacement reduces the total equipment cost while retaining the number of workstations.

The implementation of the above procedure for each equipment produces at most r efficient solutions. The number of efficient solutions is less than r if a solution found using a particular equipment is dominated by the solution found using another equipment. In such a case a dominated solution has no smaller number of workstations and no smaller total equipment cost than one existing solution.

In our branch and bound algorithm, we update the set of solutions found by the above heuristic whenever a dominating solution is found.

3.3 LOWER BOUND PROCEDURE

We calculate lower bounds for each node that cannot be fathomed by our reduction mechanisms. In each node the decision of branching or fathoming the node is decided by the lower bounds.

LB_{NS} : Lower bound on the number of workstations

LB_{TC} : Lower bound on the total equipment cost

If LB_{NS} = g and $LB_{TC} \geq UB(g)$ where $UB(g)$ is the best known upper bound on the total equipment cost with g workstations, we fathom the node.

The lower bounds, LB_{NS} and LB_{TC} are calculated separately as follows:

i. A lower bound on the number of workstations:

$$\left\lceil \frac{\sum_{i \in T'} t_{ik}}{C} \right\rceil$$

is a lower bound on the number of workstations for a single equipment assembly line balancing problem. If we replace t_{ik} with $\min_k \{t_{ik}\}$ then the resulting expression gives a lower bound on our problem with r equipment choices. We state the lower bound expression below:

$$LB_{NS} = \left\lceil \frac{\sum_{i \in T'} \min_k \{t_{ik}\}}{C} \right\rceil$$

ii. A lower bound on the total equipment cost:

Note that, when only equipment of type k is used, a lower bound on the number of workstation is

$$\left\lceil \frac{\sum_{i \in T'} t_{ik}}{C} \right\rceil .$$

The lower bound on the total equipment cost when only equipment k is used, becomes

$$LB_{TC_{Ek}} = \left\lceil \frac{\sum_{i \in T'} t_{ik}}{C} \right\rceil EC_k \; .$$

Hence a lower bound on the total equipment cost when only one type of equipment is used can be expressed as:

$$LB_{TC_1} = \min_k \left\{ \left\lceil \frac{\sum_{i \in T'} t_{ik}}{C} \right\rceil EC_k \right\}$$

A lower bound on the number of workstations when equipments E_k and E_l have to be used is

$$\left\lceil \frac{\sum_{i \in T'} \min \{t_{ik}, t_{il}\}}{C} \right\rceil .$$

When more than one equipment is used to for all unscheduled tasks, the lower bound is achieved under the assumption that only one workstation is equipped with the expensive equipment and the remaining workstations are equipped with the cheap equipment.

Accordingly, a lower bound on the number of total cost when equipments E_k and E_l such that $EC_k \leq EC_l$, have to be used is

$$LB_{TC_{Ekl}} = \left\{ \left\lceil \frac{\sum_{i \in T'} \min \{t_{ik}, t_{il}\}}{C} \right\rceil - 1 \right\} EC_k + EC_l \; .$$

A lower bound on the total cost when only two types of equipments have to be used, can be found by enumerating all combinations with two equipment types. Assume

SC_{r_p} is the set of equipments with p number of equipment combinations when there is r equipment alternatives, then when r = 3, $SC_{3_2} = \{(1,2),(1,3),(2,3)\}$ and when r = 5, $SC_{5_2} = \{(1,2),(1,3),(1,4),(1,5),(2,3),(1,2),(2,4),(2,5),(3,4),(3,5),(4,5)\}$. The associated lower bound is then $LB_{TC_2} = \min_{(k,l)\in SC_{r_2}} \{LB_{TC_{Ekl}}\}$.

A lower bound on the number of workstations when r = 3 and all the three equipments have to be used is

$$\left\lceil \frac{\sum_{i\in T'} \min\{t_{i1},t_{i2},t_{i3}\}}{C} \right\rceil.$$

When three equipments have to be used, to guarantee a lower bound we assume that one station is equipped with E_3, i.e., the third cheapest equipment, one station is equipped with E_2, i.e., the second cheapest equipment and all the remaining stations with E_1, i.e. the cheapest equipment.

Accordingly, a lower bound on the associated total cost is

$$LB_{TC_3} = \left\{ \left\lceil \frac{\sum_{i\in T'} \min\{t_{i1},t_{i2},t_{i3}\}}{C} \right\rceil - 2 \right\} EC_1 + EC_2 + EC_3.$$

An overall lower bound for the total equipment cost when r = 3 is then

$$LB_{TC} = \min\{LB_{TC_1}, LB_{TC_2}, LB_{TC_3}\}.$$

When r > 3 then enumerating all subsets of different size may be very time consuming, hence we calculate the lower bound only considering the subsets of number of equipment alternatives 1, 2 and 3. Our lower bound is

$$LB_{TC_1} = \min_k \left\{ \left\lceil \frac{\sum_{i \in T'} t_{ik}}{C} \right\rceil EC_k \right\}$$

$$LB_{TC_2} = \min_{(k,l) \in SC_{r_2}} \left\{ \left\{ \left\lceil \frac{\sum_{i \in T'} \min\{t_{ik}, t_{il}\}}{C} \right\rceil - 1 \right\} EC_k + EC_l \right\}$$

where $EC_k < EC_l$.

$$LB_{TC_3} = \min_{(k,l,s) \in SC_{r_3}} \left\{ \left\{ \left\lceil \frac{\sum_{i \in T'} \min\{t_{ik}, t_{il}, t_{is}\}}{C} \right\rceil - 2 \right\} EC_k + EC_l + EC_s \right\}$$

where $EC_k < EC_l$ and $EC_k < EC_s$.

An overall lower bound for general r is $LB_{TC} = \min\{LB_{TC_1}, LB_{TC_2}, LB_{TC_3}\}$.

In our branch and bound algorithm, in order to reduce computational time of the lower bound, we developed a procedure in which the cost elements in the expression are checked sequentially. First of all, we check whether the minimum task times of all the unscheduled tasks correspond to the cheapest equipment E_1. If this is the case, then the lower bound is equal to LB_{TC_1} and there is no need to calculate LB_{TC_2} and LB_{TC_3}. If not, the condition is checked for the equipment pair E_1 and E_2 and the lower bounds with the expensive equipment cases are not calculated.

When the number of equipments is 3, in order to increase the efficiency of the lower bound computations, we proceed as follows:

- If $\min_k \{t_{ik}\} = t_{i1} \ \ \forall i \in T'$ then LB_{TC} is simply

$$\left\lceil \frac{\sum_{i\in T'} t_{i1}}{C} \right\rceil EC_1 .$$

- If $\min_k \{t_{ik}\} = \min\{t_{i1}, t_{i2}\} \ \ \forall i \in T'$ then LB_{TC} is simply

$$\min\left\{ \left\lceil \frac{\sum_{i\in T'} t_{i1}}{C} \right\rceil EC_1, \left\{ \left\lceil \frac{\sum_{i\in T'} \min\{t_{i1}, t_{i2}\}}{C} \right\rceil - 1 \right\} EC_1 + EC_2 \right\}.$$

- If $\min_k \{t_{ik}\} = \min\{t_{i1}, t_{i3}\} \ \ \forall i \in T'$ then LB_{TC} is simply

$$\min\left\{ \left\lceil \frac{\sum_{i\in T'} t_{i1}}{C} \right\rceil EC_1, \left\lceil \frac{\sum_{i\in T'} t_{i2}}{C} \right\rceil EC_2, \left\{ \left\lceil \frac{\sum_{i\in T'} \min\{t_{i1}, t_{i2}\}}{C} \right\rceil - 1 \right\} EC_1 + EC_2, \right.$$

$$\left. \left\{ \left\lceil \frac{\sum_{i\in T'} \min\{t_{i1}, t_{i3}\}}{C} \right\rceil - 1 \right\} EC_1 + EC_3 \right\}.$$

CHAPTER 4

COMPUTATIONAL RESULTS

In this chapter, we present the results of our experiments to investigate the performances of the branch and bound algorithm and the effects of certain parameter values on the performance. Firstly, the problem generation scheme is defined. Then performance measures are stated and finally the results of the computational runs are discussed.

4.1 PROBLEM GENERATION SCHEME

We take a number of problems from the open literature. Armin SCHOLL and Robert KLEIN present benchmark data sets for SALBP at the web site http://www.assembly-line-balancing.de/. The data sets of the problems, which have been used since early 1900s, are comprehensively described.

Since our concern is not simple assembly line balancing problem (SALBP), some additional data are generated. Our model necessitates task times for each equipment, but the SALBP has one equipment for each task. We generate the task times of each problem from the uniform distribution between the minimum task time and the maximum task time. We let the original task time of the SALBP be the task time of the first equipment.

The following table gives the characteristics of the problems used. The task times and the precedence relations of the problems are given in Appendix A.

Table 4.1 Characteristics of the test problems

Problem Set	Name	n	Min. Task Time	Max. Task Time
1	Mertens	7	1	6
2	Bowman	8	3	17
3	Jaeschke	9	1	6
4	Jackson	11	1	7
5	Mansoor	11	2	45
6	Mitchell*	15	1	13
7	Mitchell**	18	1	13
8	Mitchell	21	1	13
9	Lutz1*	21	100	1400
10	Roszieg*	23	1	13
11	Roszieg	25	1	13

*, ** The reduced versions of the problems

An initial experimentation is conducted to investigate the effects of the problem parameters, i.e. the number of tasks and equipments, the equipment costs, the cycle time, the correlation between the task times and the equipment costs and the flexibility ratio. The details of these levels are presented below.

- **Problem Size, n:** The problems having n values between 7 and 25 are tested.

- **Number of Equipments, r:** r is set to 2, 4 and 5.

- **Cycle Time:** We use two values of cycle time. First we set cycle time to the maximum task time, second we set it to the 1.8*maximum task time. We refer to these versions CT1 and CT2 hereafter.

- **Correlation between the task times and the related equipment cost:** We generate two sets of task times and equipment cost combinations. In the first combination, we assign the smallest task time to the most expensive equipment whereas in the second combination, we assign the task times and equipment costs randomly.

- **Equipment Costs:** Table 4.2 reports the values used for equipment costs for each r value used in our experiments.

Table 4.2 Values of equipment costs

r		EC_1	EC_2	EC_3	EC_4	EC_5
2	Ecost1	100	200	-	-	-
	Ecost2	100	120	-	-	-
4	Ecost1	100	200	300	400	-
	Ecost2	100	100	150	200	-
5	Ecost1	100	200	300	400	500
	Ecost2	100	100	120	140	160

Note that the first combination represents high variability between the equipment costs whereas the second one represents low variability. In the second combination when r = 4 and 5, we assign same costs for the first and the second equipments.

- **Flexibility Ratio (FR):** The flexibility ratio is a measure of flexibility of the assembly line and is calculated by dividing the number of zero entries in the precedence matrix by the total number of entries. The ratio is calculated by the following expression:

$$FR = \frac{2 * (\mathit{Number\ of\ zeros\ in\ the\ precedence\ matrix})}{n * (n - 1)}$$

The value of the flexibility ratio is between of 0 and 1. Higher FR means fewer precedence relations in the matrix that leads to higher alternative solutions.

We use the precedence relations of the reported problems in the experiments. Additionally, to investigate the effect of FR on the problem difficulty we generate more dependent tasks in a network for each problem instance. The desired flexibility ratio for the high dependent case is 0.5. The number of ones in the precedence matrix that makes FR = 0.5, in the above formula, is calculated and the cells of the matrix are randomly filled by ones and zeros.

4.2 OUR PERFORMANCE MEASURES

We use the following performance measures to test the efficiency of the branch and bound algorithm and investigate the effects of the parameters.

- **Central Processing Unit (CPU) Time:** CPU times are expressed in seconds.

- **Total Number of Nodes Generated:** Total number of partial solutions evaluated by the branch and bound procedure.

- **Number of Efficient Solutions:** The number of nondominated alternatives to be presented to the decision maker.

The following measure is used to evaluate the efficiency of our ranked positional weight heuristic method.

- **Percentage Deviation of the Upper Bound from Optimal (or Best Known) Solution as a Ratio of the Optimal Solution (PD):**

$$PD = \frac{(Heuristic \;\; Solution \;\; Value \; - \; Optimal / Best \;\; Known \;\; Solution \;\; Value)}{Optimal / Best \;\; Known \;\; Solution \;\; Value} \; * \; 100$$

4.3 THE DISCUSSION OF THE RESULTS

As mentioned, we design computational experiments consisting of 11 test problems taken from literature. The effects of the performance measures and the performance of the solution procedures are investigated. Tables 4.4 through 4.12 summarize the results of these experiments.

We limit the run time of each problem instance to one hour. If the optimal solutions cannot be found in one hour, the best known solutions found until this time are recorded in our solution list.

By combining the different values of the parameters, 24 different combinations of each problem instance are formed and solved by the branch and bound algorithm. Table 4.3 illustrates the number of the problems that cannot be used to optimality within time limit of one hour.

Table 4.3 The number of unsolved instances in one hour

Problem Set	n	# of unsolved instance(s)*
9	21	1
10	23	1
11	25	3

* Out of 24 combinations

All combinations of the problems 1 through 8 are solved to optimality in one hour. Only one problem out of 24 could not be solved in one hour for problem sets 9 and 10.

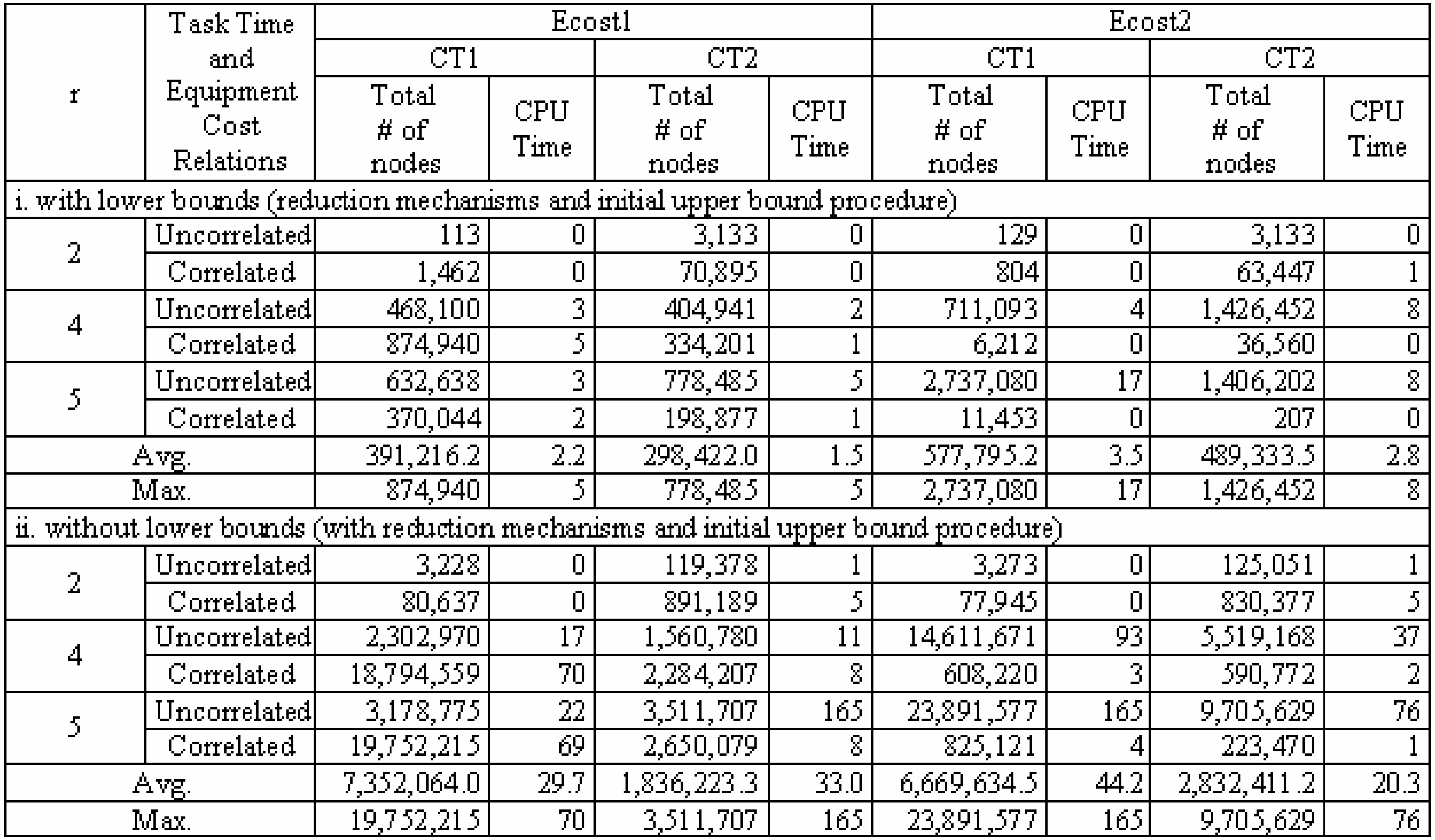
The results of the 24 versions of the problems, which include the total number of nodes generated, the CPU time and the number of efficient solutions, are given in Appendix B.

We code our algorithms in Visual C++ 6.0 version implement on a PC: Intel(R) Pentium(R) 4 CPU 3.00GHz with 512 MB RAM.

4.3.1 THE EFFECTS OF THE PROCEDURES

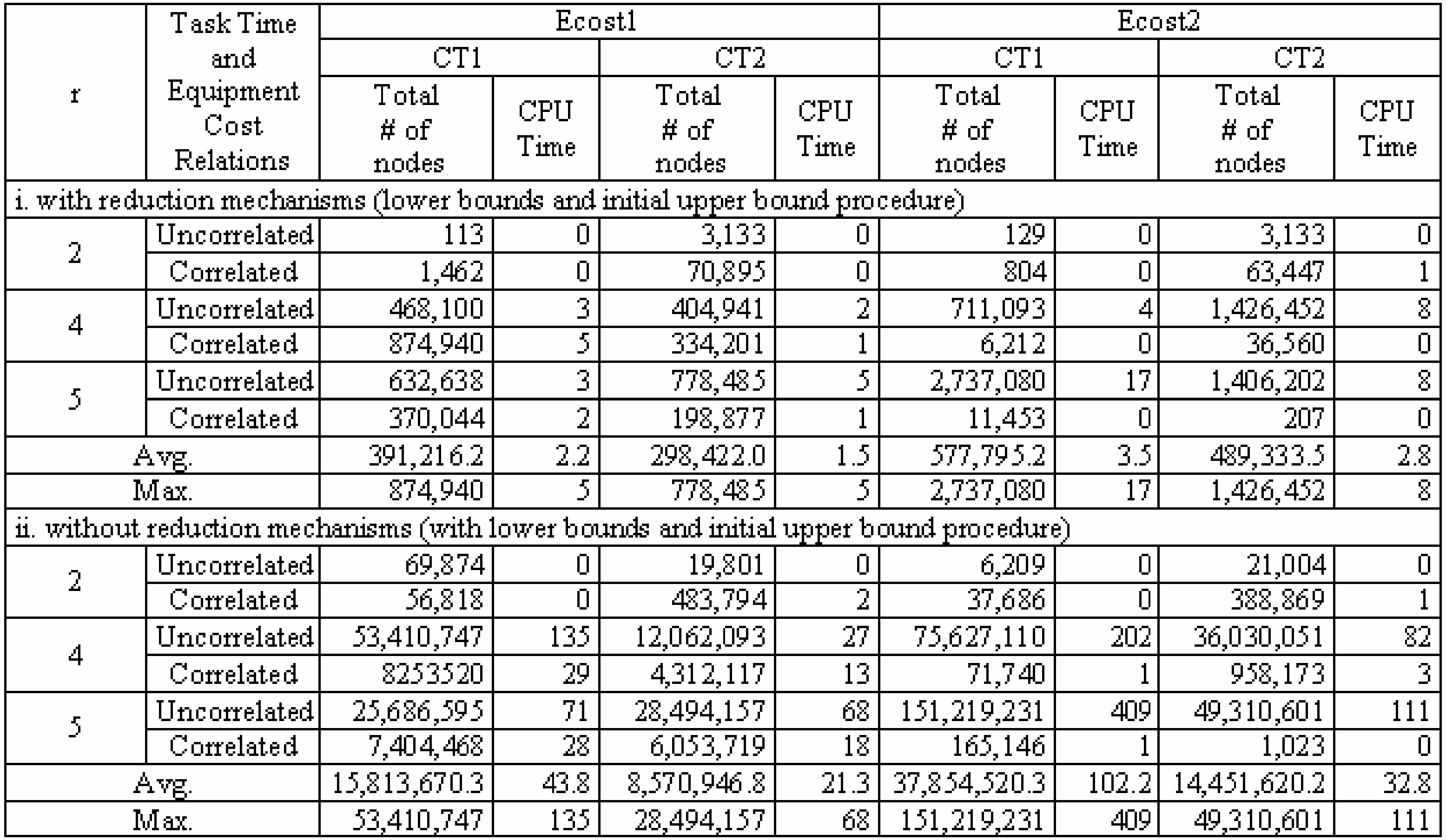
In this section, we investigate the performances of our reduction and bounding mechanisms and the efficiency of our branch and bound algorithm. For this purpose we select the problem Mitchell** that has 18 tasks and solve 24 problem instances for each of the 24 combinations. Table 4.4 through 4.6 report the results of the 24 problem instances when the procedures, lower bounds, initial upper bound procedures and reduction mechanisms, are separately removed from the branch and bound algorithm. The first parts of the tables reporting the results of our branch and bound algorithm are given for comparison.

Table 4.4 The results of the problem Mitchell** with / without lower bounds, n = 18



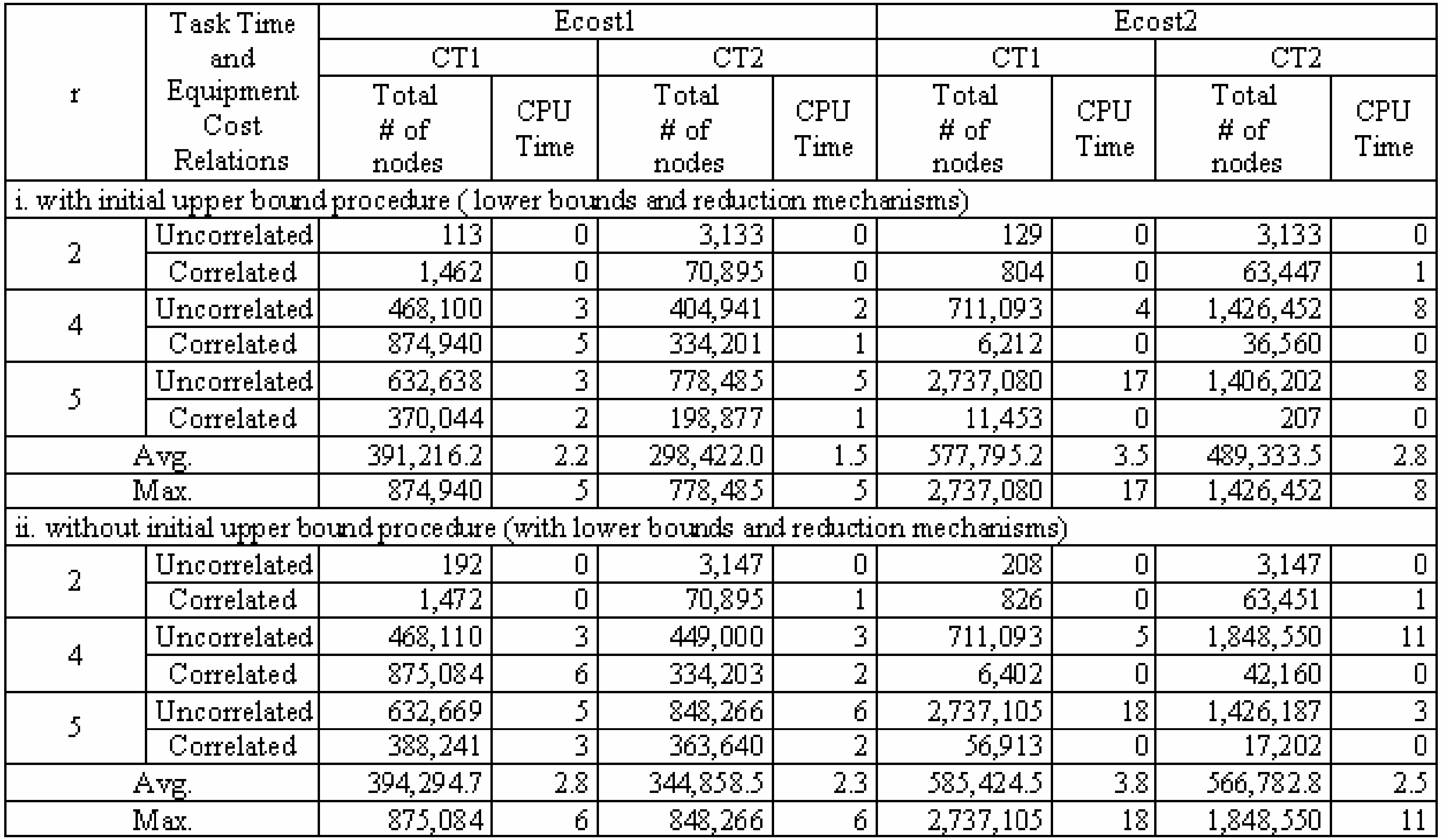
r	Task Time and Equipment Cost Relations	Ecost1				Ecost2			
		CT1		CT2		CT1		CT2	
		Total # of nodes	CPU Time	Total # of nodes	CPU Time	Total # of nodes	CPU Time	Total # of nodes	CPU Time
i. with lower bounds (reduction mechanisms and initial upper bound procedure)									
2	Uncorrelated	113	0	3,133	0	129	0	3,133	0
	Correlated	1,462	0	70,895	0	804	0	63,447	1
4	Uncorrelated	468,100	3	404,941	2	711,093	4	1,426,452	8
	Correlated	874,940	5	334,201	1	6,212	0	36,560	0
5	Uncorrelated	632,638	3	778,485	5	2,737,080	17	1,406,202	8
	Correlated	370,044	2	198,877	1	11,453	0	207	0
Avg.		391,216.2	2.2	298,422.0	1.5	577,795.2	3.5	489,333.5	2.8
Max.		874,940	5	778,485	5	2,737,080	17	1,426,452	8
ii. without lower bounds (with reduction mechanisms and initial upper bound procedure)									
2	Uncorrelated	3,228	0	119,378	1	3,273	0	125,051	1
	Correlated	80,637	0	891,189	5	77,945	0	830,377	5
4	Uncorrelated	2,302,970	17	1,560,780	11	14,611,671	93	5,519,168	37
	Correlated	18,794,559	70	2,284,207	8	608,220	3	590,772	2
5	Uncorrelated	3,178,775	22	3,511,707	165	23,891,577	165	9,705,629	76
	Correlated	19,752,215	69	2,650,079	8	825,121	4	223,470	1
Avg.		7,352,064.0	29.7	1,836,223.3	33.0	6,669,634.5	44.2	2,832,411.2	20.3
Max.		19,752,215	70	3,511,707	165	23,891,577	165	9,705,629	76

Table 4.5 The results of the problem Mitchell** with / without reduction mechanisms, n = 18



r	Task Time and Equipment Cost Relations	Ecost1				Ecost2			
		CT1		CT2		CT1		CT2	
		Total # of nodes	CPU Time	Total # of nodes	CPU Time	Total # of nodes	CPU Time	Total # of nodes	CPU Time
i. with reduction mechanisms (lower bounds and initial upper bound procedure)									
2	Uncorrelated	113	0	3,133	0	129	0	3,133	0
	Correlated	1,462	0	70,895	0	804	0	63,447	1
4	Uncorrelated	468,100	3	404,941	2	711,093	4	1,426,452	8
	Correlated	874,940	5	334,201	1	6,212	0	36,560	0
5	Uncorrelated	632,638	3	778,485	5	2,737,080	17	1,406,202	8
	Correlated	370,044	2	198,877	1	11,453	0	207	0
Avg.		391,216.2	2.2	298,422.0	1.5	577,795.2	3.5	489,333.5	2.8
Max.		874,940	5	778,485	5	2,737,080	17	1,426,452	8
ii. without reduction mechanisms (with lower bounds and initial upper bound procedure)									
2	Uncorrelated	69,874	0	19,801	0	6,209	0	21,004	0
	Correlated	56,818	0	483,794	2	37,686	0	388,869	1
4	Uncorrelated	53,410,747	135	12,062,093	27	75,627,110	202	36,030,051	82
	Correlated	8253520	29	4,312,117	13	71,740	1	958,173	3
5	Uncorrelated	25,686,595	71	28,494,157	68	151,219,231	409	49,310,601	111
	Correlated	7,404,468	28	6,053,719	18	165,146	1	1,023	0
Avg.		15,813,670.3	43.8	8,570,946.8	21.3	37,854,520.3	102.2	14,451,620.2	32.8
Max.		53,410,747	135	28,494,157	68	151,219,231	409	49,310,601	111

Table 4.6 The results of the problem Mitchell** with / without initial upper bound procedure, n = 18



r	Task Time and Equipment Cost Relations	Ecost1				Ecost2			
		CT1		CT2		CT1		CT2	
		Total # of nodes	CPU Time	Total # of nodes	CPU Time	Total # of nodes	CPU Time	Total # of nodes	CPU Time
i. with initial upper bound procedure (lower bounds and reduction mechanisms)									
2	Uncorrelated	113	0	3,133	0	129	0	3,133	0
	Correlated	1,462	0	70,895	0	804	0	63,447	1
4	Uncorrelated	468,100	3	404,941	2	711,093	4	1,426,452	8
	Correlated	874,940	5	334,201	1	6,212	0	36,560	0
5	Uncorrelated	632,638	3	778,485	5	2,737,080	17	1,406,202	8
	Correlated	370,044	2	198,877	1	11,453	0	207	0
Avg.		391,216.2	2.2	298,422.0	1.5	577,795.2	3.5	489,333.5	2.8
Max.		874,940	5	778,485	5	2,737,080	17	1,426,452	8
ii. without initial upper bound procedure (with lower bounds and reduction mechanisms)									
2	Uncorrelated	192	0	3,147	0	208	0	3,147	0
	Correlated	1,472	0	70,895	1	826	0	63,451	1
4	Uncorrelated	468,110	3	449,000	3	711,093	5	1,848,550	11
	Correlated	875,084	6	334,203	2	6,402	0	42,160	0
5	Uncorrelated	632,669	5	848,266	6	2,737,105	18	1,426,187	3
	Correlated	388,241	3	363,640	2	56,913	0	17,202	0
Avg.		394,294.7	2.8	344,858.5	2.3	585,424.5	3.8	566,782.8	2.5
Max.		875,084	6	848,266	6	2,737,105	18	1,848,550	11

We first study the power of the lower bounds on the node eliminations and the CPU reductions, and report the results for the algorithms that use lower bounds and that do not use lower bounds, in Table 4.4. We use upper bound and reduction procedures in both versions of the algorithm. As can be observed from Table 4.4, the lower bounds reduce the number of nodes, therefore CPU times, considerably. Hence the effort used to find the lower bounds is very much justified by the reduction in CPU times. For example, when r = 5, the equipment costs have high variability and the task times and the equipment costs are correlated, the average CPU times is reduced from 19,752,215 seconds to 370,044 seconds by the lower bounds.

We also report the performance of reduction mechanisms in reducing the size of the search. The associated results are tabulated in Table 4.5 for two versions of our branch and bound algorithm: using reduction mechanisms and not using reduction mechanisms. Note that the use of reduction mechanisms improves the performance of the branch and bound algorithm very significantly. The reductions are more significant when the problem sizes are larger and the reduction theorems are more likely to exist like low variable equipment costs, high cycle times and correlated task times and equipment cost cases.

We finally test the performance of our heuristic when used with in a branch and bound algorithm. Table 4.6 reports the performance of our branch and bound algorithm for Mitchell**'s set when initial upper bound used and not used cases. Note that the upper bounds slightly affect the performance. However as they are very quick, it should be incorporated. Moreover in some cases, the heuristic may return exact efficient solutions and our branch and bound algorithm may spend little effort to verify this.

In addition we also test the accuracy of the heuristic method on the 11 problem sets. We measure the performance of our initial upper bound at root node as a percentage deviation from the optimal solution and report the maximum and the average of the results of the 24 combinations in Table 4.7 for each r value.

Table 4.7 The heuristic's percentage deviations for the total equipment costs from the optimal/best known costs

Problem Set	n	r = 2		r = 4		r = 5	
		Avg.	Max.	Avg.	Max.	Avg.	Max.
1	7	17.78	66.67	25.05	60.00	18.92	33.33
2	8	3.42	16.67	4.36	80.00	4.32	80.00
3	9	15.63	25.00	14.63	25.00	14.44	25.00
4	11	26.34	125.00	40.51	95.83	32.88	70.83
5	11	9.68	33.33	10.18	20.00	9.33	36.36
6	15	5.18	20.00	40.22	100.00	35.61	100.00
7	18	2.86	9.09	24.56	67.50	23.80	67.50
8	21	4.61	12.50	8.08	22.22	5.24	15.38
9	21	34.09	100.00	41.81	100.00	39.61	100.00
10	23	21.43	21.43	22.92	28.57	27.97	48.40
11	25	16.19	37.04	27.53	80.00	24.61	68.00

As can be observed from the table most of the average deviations are below 50% and the deviations deteriorate as n gets larger. Note that when n = 18 and n = 21, the average deviations are 2.86 and 34.09 respectively for r = 2. The effect of r is not very significant on the performance. Moreover, for some problem sets when the number of tasks and the number of equipments are close the results significantly vary as can be observed from the table. This variability can be attributed to the random effect.

Our heuristic is a simple rule that returns the solution in negligible time, therefore the solution times are not reported.

4.3.2 THE EFFECTS OF THE EXPERIMENTAL PARAMETERS

In this section, we report on the performance of our branch and bound algorithm using the measures discussed in the previous section. The results of our computational experiments presented in Tables 4.8 through 4.12 have revealed that when all the parameter combinations are fixed, as the number of tasks, n, increase, the total number of nodes and CPU time increase exponentially. This effect is expected since the number of tasks influences the number of branches and the depth of the tree. The increase in the difficulty is more significant when the number of tasks is greater than 20.

The number of equipment alternatives also has a direct effect on the total number of nodes and CPU time. The size of the tree enlarges considerably by the increase in the number of equipments. Tables 4.8 though 4.12 present the results of the problem instances for all the equipment alternatives. When the number of equipments is 2 the overall average of all scores of CPU time is 0.6 seconds and when r becomes 4, the average CPU time increases to 21.3 seconds.

We observe that the number of equipments is one of the dominant factors that affects the difficulty of the problem. Moreover, for some problem instances an optimal solution cannot be obtained in one hour when there are 5 equipments.

We aim to investigate the effects of the task number and the number of equipments on the number of efficient solutions. Table 4.8 summarizes the maximum and the average number of efficient solutions of the experimental problems.

Table 4.8 The number of efficient solutions of the 11 problem sets

			Ecost1				Ecost2			
Problem Set	n	r	CT1		CT2		CT1		CT2	
			UC	C	UC	C	UC	C	UC	C
1	7	2	1	2	1	1	1	1	1	1
		4	1	2	2	1	1	1	2	1
		5	1	3	2	2	1	1	1	1
2	8	2	1	1	2	1	1	1	2	1
		4	3	3	2	2	2	1	1	1
		5	3	4	2	2	2	1	1	1
3	9	2	1	1	1	1	1	1	1	1
		4	2	3	2	1	2	1	2	1
		5	2	5	2	2	2	1	2	1
4	11	2	1	1	2	2	1	1	1	1
		4	4	3	3	2	2	1	2	1
		5	4	1	3	1	2	1	2	1
5	11	2	1	1	2	1	1	1	2	1
		4	2	1	3	1	2	1	2	1
		5	2	1	3	1	2	1	2	1
6	15	2	2	1	1	1	2	1	1	1
		4	4	2	2	2	3	1	2	1
		5	4	2	2	2	3	1	2	1
7	18	2	1	1	2	1	1	1	2	1
		4	4	2	3	1	2	1	2	1
		5	5	2	4	1	2	1	2	1
8	21	2	1	2	1	1	1	1	1	1
		4	4	4	3	2	3	1	3	1
		5	4	4	3	2	3	1	3	1
9	21	2	1	2	1	1	1	1	1	1
		4	3	3	2	2	3	1	2	1
		5	3*	3	2	2	2	1	2	1
10	23	2	2	2	2	1	2	1	2	1
		4	4	4	2	2	3	2	2	1
		5	4*	4	3	3	3	1	3	1
11	25	2	2	1	2	1	1	1	1	1
		4	3*	3	3	2	2	1	2	1
		5	4*	5	3*	5	3	1	2	1
Overall Avg.		2	1.27	1.36	1.55	1.09	1.18	1.00	1.36	1.00
		4	3.09	2.73	2.45	1.64	2.27	1.09	2.00	1.00
		5	3.27	3.09	2.64	2.09	2.27	1.00	2.00	1.00
Overall Max.		2	2	2	2	2	2	1	2	1
		4	4	4	3	2	3	2	3	1
		5	5	5	4	5	3	1	3	1

*Approximate solutions

It can be seen from the table that as the number of the equipment alternatives increases the average number of efficient solutions increases. When the number of equipments is 2, the maximum of the averages is 1.55. This value becomes 3.27 when the number of equipment alternatives increases to 5. When $n \geq 21$, the efficient solutions are approximate for some instances, as they could not be solved to optimality. Note that the effect of the number of tasks on the number of efficient solutions is similar to the number of equipments. Hence, we can conclude that the number of efficient solutions increases by the increase in the problem size. This is due to the fact that as the number of equipments and/or tasks increases the probability of the alternative solutions increases.

Moreover, the effects of the equipment cost, cycle time and the correlation between the task times and the equipments costs on the number of efficient solutions can be observed from the table. The number of efficient solutions is smaller when the equipments costs are closer to each other (Ecost2). The same effect is observed when the cycle time is higher, i.e. equal to 1.8*maximum task time, and the correlated case of the equipment costs and the task times. When the cycle time is higher the number of efficient solutions is lower, as fewer alternatives exist when CT is large. Moreover the number of workstation is smaller when CT is higher which narrows the efficient solution range. From these effects it can be concluded that as the problem gets harder the number of alternative lines and the number of efficient solutions increases.

We next investigate the effect of the cycle time on the efficiency of the branch and bound algorithm and find that the size of the tree is very sensitive to the cycle time. Table 4.9 reports the average and the maximum CPU times and the number of performance measures. We observe significant differences in the performances between the two types: CT1 (maximum task time) and CT2 (1,8*maximum task time).

Table 4.9 The effect of the cycle time on the performance of the branch and bound algorithm

Problem Set	n	r	CT1				CT2			
			Total # of nodes		CPU Time		Total # of nodes		CPU Time	
			Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.
1	7	2	76.3	90	0.0	0	20.5	35	0.0	0
		4	260.3	419	0.0	0	388.5	592	0.0	0
		5	609.3	1,362	0.0	0	736.3	1,293	0.0	0
2	8	2	51.5	63	0.0	0	11.5	14	0.0	0
		4	665.8	1,417	0.0	0	153.3	317	0.0	0
		5	912.8	2,295	0.0	0	422.0	1,312	0.0	0
3	9	2	125.0	165	0.0	0	97.8	134	0.0	0
		4	732.3	1,934	0.0	0	1,841.5	6,891	0.0	0
		5	1,120.3	3,380	0.0	0	2,443.8	9,280	0.0	0
4	11	2	2,135.8	3,356	0.0	0	587.0	940	0.0	0
		4	15,718.3	24,929	0.0	0	11,767.5	26,131	0.0	0
		5	31,057.3	50,565	0.0	0	21,369.8	30,231	0.0	0
5	11	2	1,265.5	2,470	0.0	0	878.8	2,217	0.0	0
		4	7,248.3	12,486	0.0	0	3,389.0	12,885	0.0	0
		5	10,635.3	20,507	0.0	0	14,653.5	36,793	0.0	0
6	15	2	2,886.5	7,724	0.0	0	165.3	355	0.0	0
		4	20,456.0	53,407	0.0	0	4,974.0	10,691	0.0	0
		5	51,439.5	163,190	0.3	1	18,877.3	44,633	0.0	0
7	18	2	627.0	1,462	0.0	0	35,152.0	70,895	0.3	1
		4	515,086.3	874,940	3.0	5	550,538.5	1,426,452	2.8	8
		5	937,803.8	2,737,080	5.5	17	595,942.8	1,406,202	3.5	8
8	21	2	60,414.8	213,458	0.5	2	162,275.8	486,102	1.3	4
		4	19,300,238.8	70,053,159	135.8	489	3,778,598.8	8,272,600	21.0	43
		5	105,333,958.8	213,486,645	410.3	1,450	11,690,815.0	25,538,382	63.8	125
9	21	2	182,585.5	393,296	1.8	4	65,705.8	163,014	0.5	1
		4	45,026,998.0	172,107,570	393.3	1,496	4,434,785.0	7,912,946	33.5	63
		5	110,546,195.0	425,161,312	940.0	3,600	11,876,148.5	35,136,082	72.5	193
10	23	2	586,095.0	1,248,604	4.8	10	175,750.0	393,045	1.3	3
		4	47,221,401.3	177,987,861	310.0	1,155	12,814,803.3	49,087,476	66.3	251
		5	140,295,668.5	541,877,587	938.8	3,600	46,531,170.5	176,843,884	242.3	908
11	25	2	216,700.8	478,568	1.8	4	1,296,881.8	2,705,577	10.8	23
		4	139,708,836.0	508,000,000	1,005.5	3,600	63,785,295.3	212,804,724	350.3	1,162
		5	143,569,115.8	564,000,000	920.0	3,600	177,393,259.5	660,000,000	974.3	3,600

As can be observed from the table, for almost all problem combinations, the instances are harder to solve when the cycle time is smaller, i.e., CT1 case. Note that for this case, we have higher number of workstations as fewer tasks can fit to a particular workstation. This leads to a higher number of evaluations in our branch and bound tree, as fewer numbers of alternatives would be ignored by our precedence theorems that look for the fittable tasks of the current workstation. As can be observed from the table, the differences between the performances of the two cycle time values become more pronounced as the number of tasks and/or number of equipments increase.

The effects of the correlation between the task times and the related equipment costs are also analyzed and the results are given in Table 4.10.

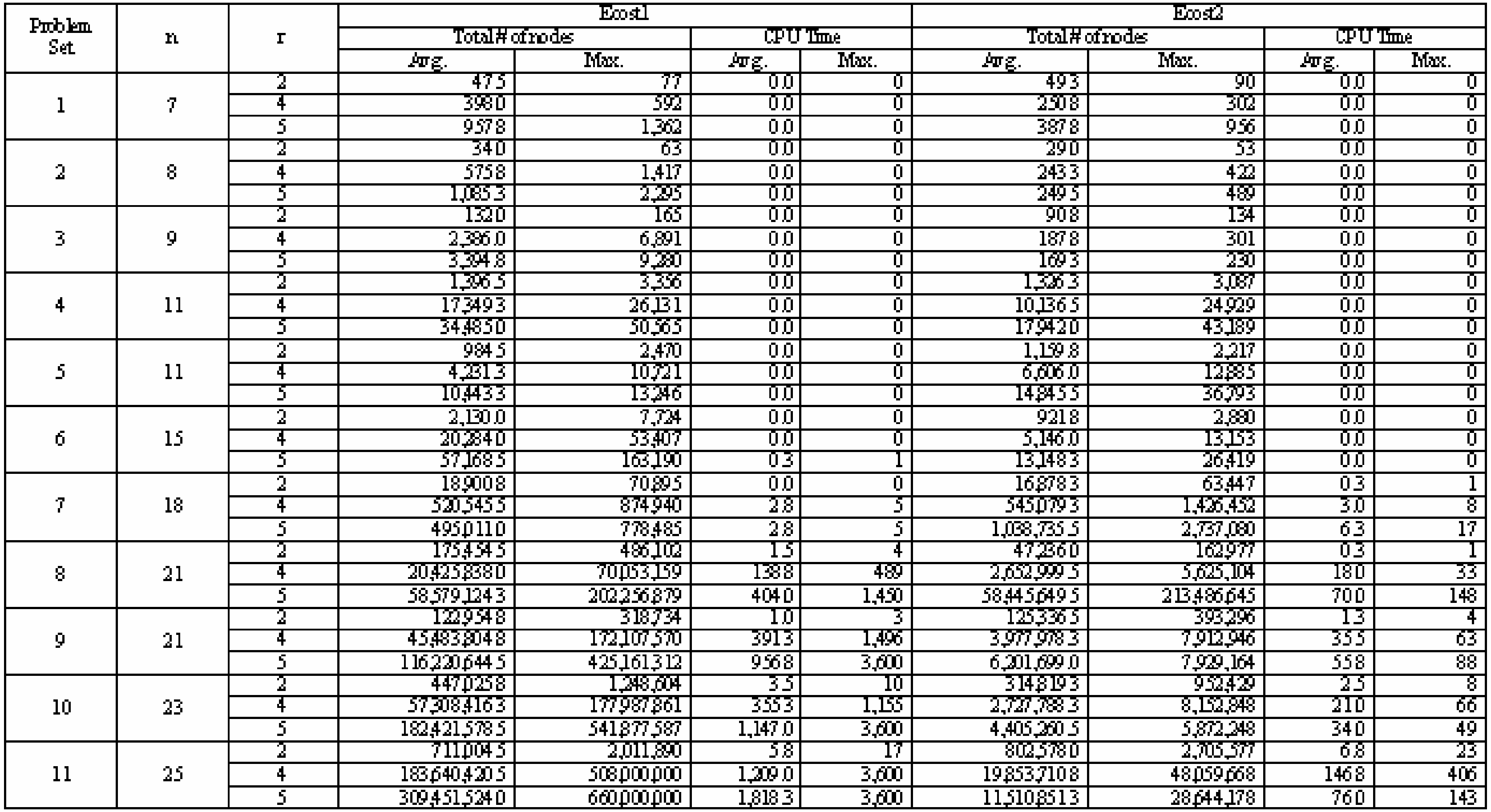
Table 4.10 The effect of the correlation between the task times and the related equipment costs on the performance of the branch and bound algorithm

Problem Set	n	r	Uncorrelated				Correlated			
			Total # of nodes		CPU Time		Total # of nodes		CPU Time	
			Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.
1	7	2	59.3	90	0.0	0	37.5	72	0.0	0
		4	346.8	419	0.0	0	302.0	592	0.0	0
		5	609.3	1,362	0.0	0	736.3	1,293	0.0	0
2	8	2	36.0	63	0.0	0	27.0	49	0.0	0
		4	299.0	515	0.0	0	520.0	1,417	0.0	0
		5	912.8	2,295	0.0	0	422.0	1,312	0.0	0
3	9	2	134.8	165	0.0	0	88.0	142	0.0	0
		4	280.8	489	0.0	0	2,293.0	6,891	0.0	0
		5	1,120.3	3,380	0.0	0	2,443.8	9,280	0.0	0
4	11	2	676.3	1,052	0.0	0	2,046.5	3,356	0.0	0
		4	14,722.3	24,929	0.0	0	12,763.5	26,131	0.0	0
		5	31,057.3	50,565	0.0	0	21,369.8	30,231	0.0	0
5	11	2	1,106.5	2,217	0.0	0	1,037.8	2,470	0.0	0
		4	7,743.8	12,885	0.0	0	3,093.5	10,721	0.0	0
		5	10,635.3	20,507	0.0	0	14,653.5	36,793	0.0	0
6	15	2	294.3	622	0.0	0	2,757.5	7,724	0.0	0
		4	8,382.8	13,153	0.0	0	16,847.3	53,407	0.0	0
		5	51,439.5	163,190	0.3	1	18,877.3	44,633	0.0	0
7	18	2	1,627.0	3,133	0.0	0	34,152.0	70,895	0.3	1
		4	752,646.5	1,426,452	4.3	8	312,978.3	874,940	1.5	5
		5	937,803.8	2,737,080	5.5	17	595,942.8	1,406,202	3.5	8
8	21	2	3,506.5	11,756	0.0	0	219,184.0	486,102	1.8	4
		4	2,910,354.0	5,625,104	18.5	33	20,168,483.5	70,053,159	138.3	489
		5	105,333,958.8	213,486,645	410.3	1,450	11,690,815.0	25,538,382	63.8	125
9	21	2	5,038.0	9,161	0.0	0	243,253.3	393,296	2.3	4
		4	3,483,572.5	7,912,946	28.0	63	45,978,210.5	172,107,570	398.8	1,496
		5	110,546,195.0	425,161,312	940.0	3,600	11,876,148.5	35,136,082	72.5	193
10	23	2	100,993.8	131,605	0.8	1	660,851.3	1,248,604	5.3	10
		4	919,638.5	1,634,084	6.3	12	59,116,546.0	177,987,861	370.0	1,155
		5	140,295,668.5	541,877,587	938.8	3,600	46,531,170.5	176,843,884	242.3	908
11	25	2	196,608.0	279,033	1.3	2	1,316,974.5	2,705,577	11.3	23
		4	7,938,263.5	17,405,181	42.0	90	195,555,867.8	508,000,000	1,313.8	3,600
		5	11,143,689.8	28,644,178	57.3	143	309,818,685.5	660,000,000	1,837.0	3,600

The average and the maximum of the nodes generated and the CPU time become larger when the correlation of task times and equipment costs increases. In addition, the difference between the performances of the two types increases as the number of the equipments increases. This is due to the following two reasons. Firstly, the correlation reduces the differences between the equipments and causes more alternative lines. The second reason is that the probability of the elimination due to our optimality properties is less in the correlated case. Our properties eliminate more nodes when the equipments costs and task times are uncorrelated which in turn improves the efficiency of the branch and bound algorithm.

We also analyze the effect of equipment costs on the performance of the branch and bound algorithm and reported the results in Table 4.11.

Table 4.11 The effect of the equipment costs on the performance of the branch and bound algorithm



Problem Set	n	r	Ecost1				Ecost2			
			Total # of nodes		CPU Time		Total # of nodes		CPU Time	
			Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.
1	7	2	47.5	77	0.0	0	49.3	90	0.0	0
		4	398.0	592	0.0	0	250.8	302	0.0	0
		5	957.8	1,362	0.0	0	387.8	956	0.0	0
2	8	2	34.0	63	0.0	0	29.0	53	0.0	0
		4	575.8	1,417	0.0	0	243.3	422	0.0	0
		5	1,085.3	2,295	0.0	0	249.5	489	0.0	0
3	9	2	132.0	165	0.0	0	90.8	134	0.0	0
		4	2,386.0	6,891	0.0	0	187.8	301	0.0	0
		5	3,394.8	9,280	0.0	0	169.3	230	0.0	0
4	11	2	1,396.5	3,356	0.0	0	1,326.3	3,087	0.0	0
		4	17,349.3	26,131	0.0	0	10,136.5	24,929	0.0	0
		5	34,485.0	50,565	0.0	0	17,942.0	43,189	0.0	0
5	11	2	984.5	2,470	0.0	0	1,159.8	2,217	0.0	0
		4	4,231.3	10,721	0.0	0	6,606.0	12,885	0.0	0
		5	10,443.3	13,246	0.0	0	14,845.5	36,793	0.0	0
6	15	2	2,130.0	7,724	0.0	0	921.8	2,880	0.0	0
		4	20,284.0	53,407	0.0	0	5,146.0	13,153	0.0	0
		5	57,168.5	163,190	0.3	1	13,148.3	26,419	0.0	0
7	18	2	18,900.8	70,895	0.0	0	16,878.3	63,447	0.3	1
		4	520,545.5	874,940	2.8	5	545,079.3	1,426,452	3.0	8
		5	495,011.0	778,485	2.8	5	1,038,735.5	2,737,080	6.3	17
8	21	2	175,454.5	486,102	1.5	4	47,236.0	162,977	0.3	1
		4	20,425,838.0	70,053,159	138.8	489	2,652,999.5	5,625,104	18.0	33
		5	58,579,124.3	202,256,879	404.0	1,450	58,445,649.5	213,486,645	70.0	148
9	21	2	122,954.8	318,734	1.0	3	125,336.5	393,296	1.3	4
		4	45,483,804.8	172,107,570	391.3	1,496	3,977,978.3	7,912,946	35.5	63
		5	116,220,644.5	425,161,312	956.8	3,600	6,201,699.0	7,929,164	55.8	88
10	23	2	447,025.8	1,248,604	3.5	10	314,819.3	952,429	2.5	8
		4	57,308,416.3	177,987,861	355.3	1,155	2,727,788.3	8,152,848	21.0	66
		5	182,421,578.5	541,877,587	1,147.0	3,600	4,405,260.5	5,872,248	34.0	49
11	25	2	711,004.5	2,011,890	5.8	17	802,578.0	2,705,577	6.8	23
		4	183,640,420.5	508,000,000	1,209.0	3,600	19,853,710.8	48,059,668	146.8	406
		5	309,451,524.0	660,000,000	1,818.3	3,600	11,510,851.3	28,644,178	76.0	143

Table 4.11 shows that the average and the maximum of the number of nodes and the CPU times decrease as the costs of the equipments become closer. This result is expected, as the costs become closer, more nodes are eliminated due to the reduction mechanisms which in turn reduces the CPU times. We also observe from the table that as r increases the difference between the performance measures becomes more significant.

Our results reported on Table 4.11 have revealed that the variability of the equipment costs is a dominant factor that affects the performance. As can be observed from the table when the costs are more variable, the problems are harder to solve. This is due to the fact that when the equipment costs are closer, the partial solutions are similar and many of these solutions can be eliminated by our reduction mechanisms. There are some exceptions where the high variability case gives better solutions, like problem set 7 with r = 4 and 5. These results can be explained by random effect and/or power of heuristics used as initial upper bound.

We further analyze the impact of the flexibility ratio on the Mitchell's problem, i.e. problem set 8 in our experiments. The precedence relation of the problem is already given in the literature and the flexibility ratio of the matrix is found to be 0.87 by the formula given in the previous section. In order to observe the effect of the flexibility ratio, the data set of the problem is combined by the new precedence structure having a flexibility ratio of 0.5. The precedence relation for the desired FR value of 0.5 is also given in Appendix A. Table 4.12 illustrates the performance measures of all the versions of the Mitchell's problem.

Table 4.12 The effect of the flexibility ratio on the problem set named Mitchell

r	FR	Task Time and Equipment Cost Relations	Ecost1						Ecost2					
			CT1			CT2			CT1			CT2		
			Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.
2	0.87	Uncorrelated	2,246	0	1	12	0	1	11,756	0	1	12	0	1
		Correlated	213,458	2	2	486,102	4	1	14,199	0	1	162,977	1	1
	0.50	Uncorrelated	519	0	1	169	0	1	511	0	1	169	0	1
		Correlated	32,008	1	1	17,214	1	1	8,581	0	1	8,292	0	1
4	0.87	Uncorrelated	2,453,285	18	4	924,308	5	3	2,638,719	18	3	5,625,104	33	3
		Correlated	70,053,159	489	4	8,272,600	43	2	2,055,792	18	1	292,383	3	1
	0.50	Uncorrelated	85,707	1	4	117,992	1	3	133,053	1	3	187,892	2	3
		Correlated	3,145,802	35	3	215,767	2	2	77,795	1	1	18,948	0	1
5	0.87	Uncorrelated	3,768,531	25	4	2,752,705	16	3	213,486,645	148	3	18,285,313	111	3
		Correlated	202,256,879	1,450	4	25,538,382	125	2	1,823,780	18	1	186,860	3	1
	0.50	Uncorrelated	246,092	3	5	320,373	3	3	518,298	5	4	684,571	6	3
		Correlated	3,063,096	35	4	298,114	3	3	58,652	1	1	16,459	1	1

As can be observed from Table 4.12, when there are more precedence relations, i.e., when the FR = 0.5, the performance of the algorithm is better for all problem combinations. This is due to the fact that when FR = 0.5, there are less tasks that can fit to the current workstations, i.e., smaller number of alternative solutions. Hence the precedence relations are more powerful in eliminating the partial solutions. The difference in the performance between FR = 0.5 and 0.87 is more significant when the number of equipments are higher and the cycle times are smaller.

It should be noted that, when heuristic procedure produces very high quality solutions, the optimal solutions can be found very easily regardless of the characteristics of the instance. For example when the cycle time is higher with r = 2 and FR = 0.87 then solutions are unexpectedly found quicker. This due to the fact that the heuristic method finds exact efficient solutions and branch and bound algorithm makes a small effort to verify this.

Moreover, an expected pattern cannot be observed easily for some problem instances. This is due to the excellent performance of the initial upper bound procedure or just due to randomness.

CHAPTER 5

CONCLUSIONS

In this thesis, we develop an exact algorithm for an assembly line balancing problem with equipment selection decisions. Two objectives are considered: minimizing the total equipment cost and the number of workstations. Our aim is to choose the type of the equipment(s) in every workstation and determine the assignment of the tasks to each workstation and equipment type. We aim to propose a set of efficient solutions and leave the choice of the best solution to the decision maker's preferences. A branch and bound algorithm is developed whose efficiency is increased with some dominance rules and powerful lower bounds. Moreover, modified ranked positional weight heuristic method is used as initial upper bound. We find that our algorithm is capable of solving the problem instances with up to 25 tasks and 5 equipments.

A set of experiments is conducted to investigate the effects of the parameters on the problem difficulty. The results show that the important parameters that affect the performance measures are the number of tasks and the number of equipments. When keeping all the other parameters fixed, the increase of the number of equipment alternatives increases the total number of nodes and the CPU time considerably. As the number of tasks increases the average number of nodes increases exponentially and the increase on the CPU time is more pronounced when n is large.

We also found that the types of the equipment costs, cycle time, the correlation between task times and equipment costs and the flexibility ratio are other factors that affect the difficulty of the problem. The hardest problem instances are the ones having small cycle time, uncorrelated task times and equipment costs, high flexibility ratio and close equipment costs.

The mechanisms used throughout the study, in particular the node elimination procedures and lower bounding schemes are found to be very effective in reducing the size of the search. The heuristic used to find an initial feasible solution is also found to be effective, but not as significant as the other mechanisms.

There are a number of further research areas, the most noteworthy of which are discussed below.

- A heuristic procedure for solving larger size of problems may be designed. Some local search procedures can be used to improve the performance of the heuristic.

- We consider deterministic task times. One extension might be to assume stochastic task times and make assignments to workstations in such a way that the probability of exceeding the cycle time is less than a required level.

- Paralleling of workstations and tasks may be studied to improve the line efficiency.

- We assume that the preference of the decision maker or theoretically the objective function is not known. The study can be extended to the case with known objective function.

- We select single equipment to perform each task from a specified equipment set. Other extensions might be to select a number of equipments for each task. Practically, the processing of a task may require several tools (equipments).

REFERENCES

Baybars, I., 1986. ‘A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem’, Management Science, 32, 8, 909-932.

Becker, C., and Scholl, A. 2003. ‘A survey on problems and methods in generalized assembly line balancing’, Friedrich-Schiller-UniversitaetJena.

Bukchin, J. and Tzur, M., 2000. ‘Design of flexible assembly line to minimize equipment cost’, IIE Transections, 32, 585-598.

Ghosh, S. and Gagnon, R.J., 1989. ‘A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems’, International Journal of Production Research, 27, 637-670.

Graves, S.C. and Holmes Redfield, C., 1988. ‘Equipment selection and task assignment for multiproduct assembly system design’, The International Journal of Flexible Manufacturing Systems, 1, 31-50.

Graves, S.C. and Lamar, B.W., 1983. ‘An integer programming procedure for assembly system design problems’, Operations Research, 31, 522 - 545.

Graves, S.C. and Whitney, D.E., 1979. ‘A mathematical programming procedure for the equipment selection and system evaluation in programmable assembly’ in Proceedings of the Eighteenth IEEE Conference on Decision and Control, Ft Lauderdale, FL., 531-536.

Scholl, A. and Becker, C., 2003. ‘State-of-the-art exact and heuristic solution procedures for simple assembly line balancing’, Friedrich-Schiller-UniversitaetJena.

Scholl, A. and Klein, R., http://www.assembly-line-balancing.de/, assembly line balancing, 2005.

Pinto, P.A., Dannenbring, D.G. and Khumawala, B.M., 1983. ‘Assembly line balancing with processing alternatives: an application’, Management Science, 29, 817-830.

Rekiek, B., Dolgui, A., Delchambre, A. and Bratcu, A., 2002b. ‘State of art of optimization methods for assembly line design’, Annual Reviews in Control 26, 163-174.

Rubinovitz, J. and Bukchin, J., 1993. ‘RALB - a heuristic algorithm for design and balancing of robotic assembly lines’, Annals of the CIRP, 42, 497-500.

APPENDIX A

DATA SETS OF THE EXPERIMENTAL PROBLEMS

In this appendix, we give the modified versions of the problem instances taken from the literature. The parameters generated for task times are given in the tables. The precedence relations are given sets, IP, such that (i, j) exists if task i is immediate predecessor of task j.

Table A.1 The task times of Mertens's Problem, i.e., Problem Set 1

	E1	E2	E3	E4	E5
T1	1	3	5	2	6
T2	5	6	2	5	2
T3	4	3	2	3	5
T4	3	2	4	4	4
T5	5	4	5	3	5
T6	6	4	3	3	2
T7	5	6	1	6	2

Immediate Predecessor Set (IP) = {(1, 2), (1, 4), (2, 3), (2, 5), (4, 7), (5, 6)}

Table A.2 The task times of Bowman's Problem, i.e., Problem Set 2

	E1	E2	E3	E4	E5
T1	11	8	14	16	14
T2	17	11	8	7	14
T3	9	3	13	5	3
T4	5	8	16	8	12
T5	8	16	5	4	13
T6	12	11	10	9	7
T7	10	17	10	5	10
T8	3	4	17	8	15

IP Set = {(1,2), (1,4), (2,3), (2,5), (4,7), (5,6)}

Table A.3 The task times of Jaeschke's Problem, i.e., Problem Set 3

	E1	E2	E3	E4	E5
T1	5	4	6	2	4
T2	3	2	6	4	5
T3	4	2	5	3	4
T4	5	1	3	6	4
T5	4	4	3	3	5
T6	5	2	4	3	3
T7	1	2	4	6	3
T8	4	3	3	7	4
T9	6	3	2	3	4

IP Set = {(1,2), (2,3), (2,4), (3,5), (3,6), (4,6), (5,7), (6,8)}

Table A.4 The task times of Jackson's Problem, i.e., Problem Set 4

	E1	E2	E3	E4	E5
T1	6	7	6	2	5
T2	2	2	5	2	4
T3	5	1	5	1	6
T4	7	4	1	4	4
T5	1	4	4	2	7
T6	2	1	2	2	4
T7	3	6	4	5	5
T8	6	6	5	5	2
T9	5	7	6	1	5
T10	5	2	3	2	7
T11	4	3	7	1	6

IP Set = {(1,2), (1,3), (1,4), (1,5), (2,6), (3,7), (4,7), (5,7), (6,8), (7,9), (8,10), (9,11), (10,11)}

Table A.5 The task times of Mansoor's Problem, i.e., Problem Set 5

	E1	E2	E3	E4	E5
T1	4	7	13	33	12
T2	38	10	12	15	11
T3	45	18	14	6	11
T4	12	37	11	42	3
T5	10	32	44	19	12
T6	8	37	13	15	19
T7	12	37	5	19	41
T8	10	32	23	2	19
T9	2	28	2	12	32
T10	10	20	37	20	22
T11	34	13	13	6	32

IP Set = {(1,4), (2,4), (2,5), (3,11), (4,6), (5,7), (6,8), (7,9), (8,10), (9,10), (10,11)}

Table A.6 The task times of Mitchell*'s Problem, i.e., Problem Set 6

	E1	E2	E3	E4	E5
T1	4	4	5	4	9
T2	3	5	11	5	7
T3	9	2	5	4	11
T4	5	11	8	2	12
T5	9	6	6	1	7
T6	4	7	1	5	12
T7	8	11	4	7	5
T8	7	7	12	2	2
T9	5	12	3	6	4
T10	1	7	9	10	2
T11	3	12	5	10	12
T12	1	1	9	13	3
T13	5	12	8	11	10
T14	3	10	1	7	8
T15	5	2	11	2	8

IP Set = {(1,2), (1,3), (3,4), (4,5), (5,6), (5,7), (6,8), (7,8), (7,14), (8,9), (9,10), (9,11), (9,12), (9,13), (10,15), (11,15), (12,15)}

Table A.7 The task times of Mitchell**'s Problem, i.e., Problem Set 7

	E1	E2	E3	E4	E5
T1	4	11	12	2	1
T2	3	11	5	4	8
T3	9	5	2	7	11
T4	5	11	11	1	2
T5	9	9	8	13	4
T6	4	4	6	7	3
T7	8	7	4	11	1
T8	7	11	13	9	9
T9	5	7	4	8	13
T10	1	4	9	4	5
T11	3	11	7	2	13
T12	1	11	2	4	10
T13	5	1	12	3	11
T14	3	5	3	13	10
T15	5	2	4	3	6
T16	3	6	11	12	12
T17	13	13	5	2	13
T18	5	12	2	1	2

IP Set = {(1,2), (1,3), (3,4), (4,5), (5,6), (5,7), (6,8), (7,8), (7,14), (8,9), (9,10), (9,11), (9,12), (9,13), (10,15), (11,15), (12,15), (13,17), (13,18), (15,16), (15,18), (16,17)}

Table A.8 The task times of Mitchell’s Problem, i.e., Problem Set 8

	E1	E2	E3	E4	E5
T1	4	7	13	4	6
T2	3	12	5	10	13
T3	9	8	10	12	12
T4	5	13	6	5	5
T5	9	8	13	3	2
T6	4	3	1	2	8
T7	8	10	2	2	10
T8	7	6	4	10	2
T9	5	9	13	11	13
T10	1	4	12	2	15
T11	3	4	7	3	13
T12	1	7	9	7	9
T13	5	13	8	2	10
T14	3	11	6	1	8
T15	5	7	7	3	7
T16	3	9	1	5	10
T17	13	11	10	7	6
T18	5	1	8	10	1
T19	2	10	10	10	5
T20	3	8	8	3	10
T21	7	4	5	8	5

IP Set = {(1,2), (1,3), (2,21), (3,4), (4,5), (4,21), (5,6), (5,7), (6,8), (7,8), (7,14), (8,9), (9,10), (9,11), (9,12), (9,13), (10,15), (11,15), (12,15), (13,17), (13,18), (14,19), (15,16), (15,18), (16,17), (17,20), (18,19)}

The following set of precedence relations are generated randomly for the Mitchell’s Problem for achieving a flexibility ratio of 0.5.

IP Set = {(1,3), (1,5), (1,9), (1,10), (1,11), (1,13), (1,14), (1,15), (1,17), (1,19), (1,21), (2,4), (2,8), (2,10), (2,11), (2,13), (2,14), (2,15), (2,17), (2,20), (3,4), (3,9), (3,11), (3,12), (3,14), (3,17), (3,19), (3,20), (4,5), (4,7), (4,9), (4,11), (4,12), (4,13), (4,14), (4,20), (5,9), (5,12), (5,13), (5,14), (5,15), (5,16), (5,18), (5,21), (6,7), (6,12), (6,13), (6,14), (6,17), (6,18), (6,19), (6,20), (6,21), (7,8), (7,14), (7,15), (7,17), (7,18), (7,19), (7,20), (7,21), (8,9), (8,10), (8,11), (8,14), (8,15), (8,16), (8,17), (8,19), (9,10), (9,11), (9,12), (9,13), (9,14), (9,15), (9,16), (9,17), (9,20), (9,21), (10,14), (10,21), (11,13), (11,14), (11,15), (11,18), (11,20), (12,14), (12,16), (12,18), (12,20), (13,14), (13,15), (13,16), (13,17), (14,15), (14,16), (15,16), (15,17), (15,18), (16,18), (16,21), (17,19), (18,21), (19,20), (19,21)}

Table A.9 The task times of Lutz1*'s Problem, i.e., Problem Set 9

	E1	E2	E3	E4	E5
T1	458	419	235	281	1141
T2	276	198	519	223	1382
T3	520	629	1242	767	1125
T4	1400	1018	596	834	1043
T5	352	370	751	908	336
T6	196	623	225	1050	128
T7	214	1327	943	272	1271
T8	456	790	279	732	1146
T9	646	681	1225	151	1094
T10	512	209	779	666	697
T11	408	620	655	464	1386
T12	262	831	483	224	1149
T13	544	1187	144	908	1345
T14	202	1318	1201	1335	119
T15	458	540	649	965	705
T16	694	281	514	897	273
T17	616	893	242	715	547
T18	678	443	766	1199	910
T19	328	176	488	244	821
T20	324	344	479	581	1010
T21	100	920	997	466	1291

IP Set = {(1,5), (2,6), (3,9), (4,15), (5,6), (6,7), (6,8), (6,9), (7,21), (8,21), (9,10), (9,11), (10,16), (11,12), (12,13), (13,14), (13,15), (14,16), (14,17), (15,17), (16,18), (17,19), (18,21), (19,20), (20,21), (23,25)

Table A.10 The task times of Roszieg*'s Problem, i.e., Problem Set 10

	E1	E2	E3	E4	E5
T1	4	10	11	3	4
T2	3	8	2	12	11
T3	9	8	11	7	3
T4	5	3	1	2	12
T5	9	5	7	3	9
T6	4	3	11	7	2
T7	8	3	9	8	3
T8	7	11	11	12	11
T9	5	3	7	10	4
T10	1	7	2	8	3
T11	3	11	12	8	1
T12	1	4	2	8	3
T13	5	6	5	4	6
T14	3	12	8	6	4
T15	5	5	6	7	12
T16	3	2	2	7	13
T17	13	10	9	4	6
T18	5	2	11	6	10
T19	2	2	5	6	9
T20	3	7	9	4	12
T21	7	10	7	6	5
T22	5	11	7	5	7
T23	3	13	6	5	8

IP Set = {(1,3), (2,3), (3,4), (4,5), (4,8), (5,6), (6,7), (6,10), (7,11), (7,12), (8,9), (8,11), (9,13), (9,10), (11,13), (12,15), (13,14), (14,16), (14,19), (14,20), (15,17), (15,22), (16,18), (17,18), (17,23), (19,22), (20,21), (21,22)}

Table A.11 The task times of Roszieg's Problem, i.e., Problem Set 11

	E1	E2	E3	E4	E5
T1	4	7	9	13	3
T2	3	9	3	11	8
T3	9	4	3	5	10
T4	5	11	10	11	10
T5	9	2	8	1	2
T6	4	4	4	9	9
T7	8	9	7	8	6
T8	7	4	7	1	12
T9	5	6	12	2	6
T10	1	2	5	3	6
T11	3	10	12	3	12
T12	1	1	6	10	1
T13	5	12	12	3	10
T14	3	9	3	11	13
T15	5	5	8	12	6
T16	3	6	12	6	13
T17	13	3	4	5	7
T18	5	4	3	6	8
T19	2	9	3	3	9
T20	3	8	11	3	5
T21	7	4	11	13	12
T22	5	7	1	3	10
T23	3	7	2	7	5
T24	8	5	11	10	9
T25	4	6	4	8	13

IP Set = {(21,22), (1,3), (21,24), (2,3), (3,4), (4,5), (4,8), (5,6), (6,7), (6,10), (7,11), (7,12), (8,9), (8,11), (9,13), (9,10), (11,13), (12,15), (13,14), (14,16), (14,19), (14,20), (15,17), (15,22), (16,18), (17,18), (17,23), (18,25), (19,22), (20,21), (20,25)}

APPENDIX B

COMPUTATIONAL RESULTS OF THE EXPERIMENTS

The detailed results of the experiments are given in the tables of Appendix B. The tables reports the performance measures such as the total number of nodes generated, the CPU time by our branch and bound algorithm and the number of the efficient solutions.

Table B.1 The results of the problems with 2 equipment alternatives

Problem Set	n	Ecost1						Ecost2					
		CT1			CT2			CT1			CT2		
		Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.
i. Uncorrelated Task Times and Equipment Costs													
1	7	77	0	1	35	0	1	90	0	1	35	0	1
2	8	63	0	1	14	0	2	53	0	1	14	0	2
3	9	165	0	1	127	0	1	113	0	1	134	0	1
4	11	1,052	0	1	238	0	2	1,048	0	1	367	0	1
5	11	361	0	1	908	0	2	940	0	1	2,217	0	2
6	15	320	0	2	121	0	1	622	0	2	114	0	1
7	18	113	0	1	3,133	0	2	129	0	1	3,133	0	2
8	21	2,246	0	1	12	0	1	11,756	0	1	12	0	1
9	21	9,161	0	1	910	0	1	9,151	0	1	930	0	1
10	23	17,431	0	2	129,023	1	2	125,916	1	2	131,605	1	2
11	25	162,533,000	1	2	191,027	1	2	153,839	1	1	279,033	2	1
ii Correlated Task Times and Equipment Costs													
1	7	72	0	2	6	0	1	66	0	1	6	0	1
2	8	49	0	1	10	0	1	41	0	1	8	0	1
3	9	142	0	1	94	0	1	80	0	1	36	0	1
4	11	3,356	0	1	940	0	2	3,087	0	1	803	0	1
5	11	2,470	0	1	199	0	1	1,291	0	1	191	0	1
6	15	7,724	0	1	355	0	1	2,880	0	1	71	0	1
7	18	1,462	0	1	70,895	0	1	804	0	1	63,447	1	1
8	21	213,458	2	2	486,102	4	1	14,199	0	1	162,977	1	1
9	21	318,734	3	2	163,014	1	1	393,296	4	1	97,969	1	1
10	23	1,248,604	10	2	393,045	3	1	952,429	8	1	49,327	0	1
11	25	478,568	4	1	2,011,890	17	1	71,863	1	1	2,705,577	23	1

Table B.2 The results of the problems with 4 equipment alternatives

Problem set	n	Ecost1						Ecost2					
		CT1			CT2			CT1			CT2		
		Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.
i. Correlated Task Times and Equipment Costs													
1	7	419	0	1	375	0	2	291	0	1	302	0	2.0
2	8	515	0	3	54	0	2	422	0	2	205	0	1.0
3	9	489	0	2	230	0	2	205	0	2	199	0	2.0
4	11	17,416	0	4	8,788	0	3	24,929	0	2	7,756	0	2.0
5	11	4,802	0	2	802	0	3	12,486	0	2	12,885	0	2.0
6	15	13,119	0	4	3,919	0	2	13,153	0	3	4,140	0	2.0
7	18	468,100	3	4	404,941	2	3	711,093	4	2	1,426,452	8	2.0
8	21	2,453,285	18	4	924,308	5	3	2,638,719	18	3	5,625,104	33	3.0
9	21	1,043,398	9	3	2,270,281	17	2	2,707,665	23	3	7,912,946	63	2.0
10	23	1,634,084	12	4	524,244	3	2	1,110,812	7	3	409,494	3	2.0
11	25	2,184,761	12	3	11,572,197	62	3	590,915	4	2	17,405,181	90	2.0
ii. Correlated Task Times and Equipment Costs													
1	7	206	0	2	592	0	1	125	0	1	285	0	1.0
2	8	1,417	0	3	317	0	2	309	0	1	37	0	1.0
3	9	1,934	0	3	6,891	0	1	301	0	1	46	0	1.0
4	11	17,062	0	3	26,131	0	2	3,466	0	1	4,395	0	1.0
5	11	10,721	0	1	600	0	1	984	0	1	69	0	1.0
6	15	53,407	0	2	10,691	0	2	2,145	0	1	1,146	0	1.0
7	18	874,940	5	2	334,201	1	1	6,212	0	1	36,560	0	1.0
8	21	70,053,159	489	4	8,272,600	43	2	2,055,792	18	1	292,383	3	1.0
9	21	172,107,570	1,496	3	6,513,970	43	2	4,249,359	45	1	1,041,943	11	1.0
10	23	177,987,861	1,155	4	49,087,476	251	2	8,152,848	66	2	1,237,999	8	1.0
11	25	508,000,000	3,600	3	212,804,724	1,162	2	48,059,668	406	1	13,359,079	87	1.0

Table B.3 The results of the problems with 5 equipment alternatives

Problem set	n	Ecost1						Ecost2					
		CT1			CT2			CT1			CT2		
		Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.	Total # of nodes	CPU Time	# of eff. soln.
i. Uncorrelated Task Times and Equipment Costs													
1	7	619	0	1	557	0	2	243	0	1	956	0	1
2	8	669	0	3	65	0	2	489	0	2	226	0	1
3	9	489	0	2	230	0	2	230	0	2	199	0	2
4	11	29,125	0	4	28,019	0	3	43,189	0	2	26,531	0	2
5	11	9,080	0	2	13,246	0	3	20,507	0	2	36,793	0	2
6	15	14,440	0	4	6,411	0	2	26,419	0	3	23,923	0	2
7	18	632,638	3	5	778,485	5	4	2,737,080	17	2	1,406,202	8	2
8	21	3,768,531	25	4	2,752,705	16	3	213,486,645	148	3	18,285,313	111	3
9	21	1,626,958	13	3	2,958,226	21	2	7,467,346	59	2	7,883,528	57	2
10	23	7,878,911	60	4	3,085,932	20	3	5,872,248	46	3	5,694,721	36	3
11	25	3,681,544	22	4	10,124,552	51	3	2,124,485	13	3	28,644,178	143	2
ii. Correlated Task Times and Equipment Costs													
1	7	1,362	0	3	1,293	0	2	213	0	1	139	0	1
2	8	2,295	0	4	1,312	0	2	198	0	1	85	0	1
3	9	3,580	0	5	9,280	0	2	182	0	1	66	0	1
4	11	50,565	0	1	30,231	0	1	1,350	0	1	698	0	1
5	11	11,525	0	1	7,922	0	1	1,429	0	1	653	0	1
6	15	163,190	1	2	44,633	0	2	1,709	0	1	542	0	1
7	18	370,044	2	2	198,877	1	1	11,453	0	1	207	0	1
8	21	202,256,879	1,450	4	25,538,382	125	2	1,823,780	18	1	186,860	3	1
9	21	425,161,312	3,600	3	35,136,082	193	2	7,929,164	88	1	1,526,758	19	1
10	23	541,877,587	3,600	4	176,843,884	908	3	5,553,928	49	1	500,145	5	1
11	25	564,000,000	3,600	5	660,000,000	3,600	5	4,470,434	45	1	10,804,308	103	1