PART SELECTION PROBLEM
IN
DISASSEMBLY SYSTEMS

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

BY

AYÇA YETERE

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 of Science.

Prof. Dr. Meral Azizoğlu
Co-Supervisor

Assoc. Prof. Dr. Levent Kandiller
Supervisor

Examining Committee Members

Prof. Dr. Nur Evin Özdemirel	(METU,IE)	____________
Assoc. Prof. Dr. Levent Kandiller	(METU,IE)	____________
Prof. Dr. Meral Azizoğlu	(METU,IE)	____________
Assist. Prof. Dr. Pelin Bayındır	(METU,IE)	____________
Dr. Tevhide Altekin	(Sabancı Uni.,FMAN)	____________

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 : Ayça YETERE

Signature :

ABSTRACT

PART SELECTION PROBLEM IN DISASSEMBLY SYSTEMS

YETERE, Ayça

Ms.Sc. Department of Industrial Engineering

Supervisor: Assoc. Prof. Dr. Levent KANDİLLER

Co-Supervisor: Prof. Dr. Meral AZİZOĞLU

January 2006, 154 pages

In this study, we consider the disassembly problem of end-of-life (EOL) products for recovering valuable parts or assemblies. All parts obtained by disassembly processes of an EOL product may not be profitable due to their high recovery costs. Our problem is to select the parts to be released and determine the associated disassembly tasks so as to maximize the total profit. We first tackle the simple part selection problem, and then introduce a time constraint for the tasks to be performed for selected parts and search for incomplete time constrained sequences. We formulate our first problem as a Mixed Integer Problem and show that the constraint set of this formulation is totally unimodular. We also provide the dual formulation of our problem and its interpretation. For time-constrained part selection problem we propose a branch-and-bound algorithm. We first develop some reduction mechanism to reduce the size of the problem. Our solution procedure is capable of solving problems with up to 94 parts and tasks.

Keywords: Selective disassembly, product recovery, branch and bound algorithm.

ÖZ

DEMONTAJ SİSTEMLERİNDE PARÇA SEÇİMİ PROBLEMİ

YETERE, Ayça
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Levent KANDİLLER
Ortak Tez Yöneticisi: Prof. Dr. Meral AZİZOĞLU

Ocak 2006, 154 sayfa

Bu çalışmada, yaşam sonu gelmiş ürünlerin içerdiği kıymetli parçaların, demontaj yolu ile geri kazanımı problemi araştırılmıştır. Yaşam sonu gelmiş ürünlerin demontajı sonucunda elde edilen parçaların bir kısmı geri kazanım maliyetlerinin yüksek olması nedeni ile karlı değildir. Bu çalışmada ele alınan problem, toplam karın en iyilenmesi amacı ile elde edilecek parçaların ve bu parçaların elde edilebilmesi için gerekli demontaj aktivitelerinin belirlenmesini içermektedir. Öncelikli olarak üstünde çalışılan problem parça seçimi probleminin en basit halidir, daha sonra gerçekleştirilebilecek aktiviteler için zaman kısıtı konularak, demontaj aktivitelerinin belirlenmesine çalışılmıştır. İlk problem karmaşık tam sayılı programlama ile modellenmiş ve bu formülasyonda kısıt matrisinin özelliği nedeni ile kısıtların gevşetilerek doğrusal programlama formülasyonuna çevirilmesi ile tam sayı sonuçların elde edilebileceği gösterilmiştir. Ayrıca bu problemin dual formülasyonu da verilmiş ve bu formülasyon yorumlanmıştır. Zaman kısıtlı parça seçimi problemi için bir dal sınır algoritması önerilmiştir. Öncelikli olarak problem boyutlarının küçültülmesi amacı ile bir azaltım algoritması geliştirilmiştir. Önerdiğimiz çözüm algoritması 94 parça ve aktiviteye kadar olan problem büyüklüklerini çözebilmektedir.

Anahtar Kelimeler: Kısmi demontaj, ürün geri kazanımı, dal sınır algoritması.

ACKNOWLEDGMENTS

I would like to express by deepest gratitude to Assoc. Prof. Dr. Levent Kandiller and Prof. Dr. Meral Azizoğlu for their supervision, patience and understanding. There were times I thought I wouldn't be able to complete this study, but with their kind support I finally succeed.

I would like to express my sincere thanks to Prof. Dr. Nur Evin Özdemirel, Assist. Prof. Dr. Pelin Bayındır and Dr. Tevhide Altekin for their precious suggestions and comments for improving this study.

I am indebted to my colleagues at ROKETSAN, especially to Sartuk Karasoy and Barlas Ortaç, for their understanding and support.

I also would like all my friends to know that, the joy they have brought with them is the very essence of life for me. But I would like to thank specially to Ayışığı Sevdik, Sibel Alumur and Azra Timur for being there.

Also I am very grateful to my aunt Nahide Dilek, she raised me and even now being far from me she always kept in touch and let me know that she loves me and believes in me.

Finally I would like to thank my parents Halide Yetere and Ali Yetere, without them nothing could have been possible. Their love, kindness and support are the greatest sources I keep nourishing and inspiring from. Thank you for providing me the best of everything for all those years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ALBP	Assembly Line Balancing Problem
B&B	Branch and Bound
CT	Cycle Time
DCG	Disassembly Constraint Graph
dfs	depth-first-search
DPM	Disassembly Precedence Matrix
DPP	Disassembly Process Plan
DSP	Disassembly Sequence Plan
ECM	Environmentally Conscious Manufacturing
EOL	End-of-Life
LB	Lower Bound
LP	Linear Programming
MRP	Material Requirement Planning
OPT	Optimum Profit
PM	Precedence Matrix
PN	Part Releasing Tasks Matrix
UB	Upper Bound
UB1	Upper Bound One
UB2	Upper Bound Two
UB3	Upper Bound Three

CHAPTER 1

INTRODUCTION

Disassembly is the process of decomposing a product into its parts or constituents. Traditionally, product disassembly was performed to service a product or repair a part. However, the disassembly of a product, in particular at the end of its useful life or after commercial returns, is becoming a common and worthwhile industrial practice (Das et al., 2000).

Disassembly processes in its simplest sense are comprised of disassembly activities, their precedence relations and the parts released. Some reasons cited by Lambert (1999), Güngör and Gupta (2001b) and Lee et al. (2001) for disassembly operations are:

- Recovery of valuable parts or subassemblies (in short supply) which are common to other products still being produced,
- recovery of pure material fractions,
- retrieval of parts or subassemblies of old products to satisfy their sudden demand,
- removal of hazardous parts,
- increasing the purity of the remainder of the product for the purpose of chemical reclamation,
- extraction of parts from the remainder of the product which can be sent to inventory for future use,
- decreasing the amount of residue to be sent to landfills,
- achieving environmentally friendly manufacturing standards (i.e., meeting the required ratio of using recycled parts to using new parts),
- maintenance and repair.

Depending on those reasons the extent of the disassembly process must be determined by considering environmental, economical and technical factors. Either way any combination of disassembly operations help reducing the environmental and ecological damages associated with product disposal while extracting the value through recovery.

Disassembly sequences can be defined as arrangement of disassembly activities, so called tasks. A disassembly sequence usually begins with a product to be disassembled and terminates in a state where the entire product (complete disassembly) or certain parts are disconnected (selective disassembly) (Lee et al., 2001).

A feasible solution to the disassembly sequencing problem should satisfy the precedence relations among the activities, in such a way that an activity can be performed only if all its predecessors are completed.

There may be various objective functions in disassembly problems. Among them, the most frequently used in the literature are (Lee et al. 2001):
(a) minimizing costs,
(b) maximizing overall profit,
(c) maximizing the number of parts to be reused,
(d) minimizing the amount of landfill waste.

In this study, we consider the disassembly of products for recovering valuable parts or subassemblies regardless of their end-of-life (EOL) recovery choice like reusing and recycling. All parts obtained by the disassembly processes of an EOL product may not be profitable due to their high disassembly costs. Our problem, in its simplest form, is to select the parts to be released and their associated disassembly activities so as to maximize the total profit.

We handle two versions of the problem; first we tackle with the simple part selection problem, and then introduce a time constraint for the tasks to be performed for selected parts. The time constraint is introduced on the total processing time of those tasks.

In our study we will consider infinite demand for every part that can be obtained from the disassembly operations. This assumption is mostly valid for material recovery cases (materials such as copper, iron, etc.). However in part recovery, where parts are extracted for reuse, demands are finite and in varying amounts. Yet we will consider a supply-driven approach, where there is no restriction on demand other than market prices of the resulting parts and materials.

We formulate our first problem as a Mixed Integer Problem and show that the constraint set of this formulation is totally unimodular whose Linear Programming (LP) relaxation yields integral solutions. We also provide the dual formulation of our problem together with its interpretation.

We next study the part selection problem with a constraint on the total time of the tasks. We first develop some reduction mechanisms to reduce the size of the problem. We then solve the reduced problem by the branch and bound algorithm and increase the efficiency of the algorithm by introducing elimination and bounding mechanisms.

Our contribution with this study is to provide reduction mechanisms for disassembly precedence graphs and a solution methodology for the part selection problem under time constraint. We also show that the part selection problem without time constraints incorporating AND precedence relations, when modeled as an integer programming problem has a totally unimodular coefficient matrix, therefore its linear programming relaxation provides all integer values for decision variables.

In the next chapter we briefly discuss the practical importance of disassembly operations by providing information on recovery processes on EOL products. The general description of the disassembly process, the sequencing problem and the solution approaches are also summarized.

In Chapter 3, we define our problems; give the mathematical programming formulations together with their underlying assumptions. We discuss the complexity statue of the problems.

In Chapter 4 the problem size reduction mechanism and the associate algorithm are presented.

In Chapter 5, we discuss the details of our branch-and-bound approach for the time constrained disassembly selection problem.

In Chapter 6, we present the results of our computational experiment and discuss the effects of the parameters on the difficulty of attaining solutions.

We conclude in Chapter 7 by pointing out the contributions of our study and the future research directions.

CHAPTER 2

LITERATURE REVIEW

In this section, we overview the product recovery process, its practical importance, definitions and related issues. Disassembly process and its application are also presented, accompanied by a thorough presentation of the disassembly sequencing problem and the related literature. A summary table of the reviewed literature and the contribution of our work are given at the end of the section.

2.1 END-OF-LIFE PRODUCT RECOVERY

Over the last decade many European countries, US and Japan have passed legislations for mandatory product recovery due to the increase of the discarded end-of-life (EOL) products without being treated in an environment friendly way (Toffel, 2002). Those regulations resulted in a new era, where manufacturers are held responsible for taking EOL products from customers and then recovering them in a proper manner dictated by the law.

The legislations imposed different sanctions on manufacturer. Government actions include disposal bans for specific products, recycled content mandates, recycling goals and product take-back requirements. Some governments even imposed purchasing programs favoring products that are reusable or that have reused content (Thierry et al., 1995). The legislations so far cover only a group of products like cars, tires, packaging, batteries and electrical and electronic equipments including house hold appliances, air conditioners, televisions, computers, printers, copiers and telephones (Lee et al., 2001, Thierry et al., 1995, Toffel, 2002).

Apart from those dictated by the law, many other product manufacturers are recovering their products. Some of those companies recover their products as a part of a non-marketing strategy to preempt or delay similar legislations, to reduce costs, to provide new revenue streams or to improve their green corporate images and their environmental performance (Toffel, 2002).

All those new regulations and public environmental demands ultimately have promoted environmentally conscious manufacturing, which, according to Güngör and Gupta (2001a), is concerned with developing methods for manufacturing new products from conceptual design to final delivery, and ultimately to the end-of-life disposal such that the environmental standards and requirements are satisfied. Therefore, manufacturers start implementing product designs which allow the reuse of parts and materials that yields an increase in the recovery values of EOL products.

Many companies contemplate the recovery of discarded products a cost deriving issue for their business. However, there could be opportunities for companies that succeed in ways of adapting current and future environment demands in their business policy. Companies could use discarded products as a valuable source of component and materials to provide cheaper supplies for use in repairs and maintenance (as spare parts) or to replace virgin components in manufacturing.

For example, BMW is selling remanufactured high-value components such as engines, starter motors and alternators as “Exchange Parts” for 50 to 70% of new prices (Thierry et al., 1995). Another example is Xerox Corporation, where they manage to eliminate million of dollars in annual logistical, inventory and raw-material costs by disassembling its EOL photocopiers and then cleaning, sorting, and repairing components for remanufacturing into new models and by recycling residual materials (Toffel, 2002).

2.1.1 RECOVERY PROCESS

Recovery is the organization and execution of all the activities associated with the reuse of all discarded products, components and materials. There are two main drivers for product recovery (1) government regulations, which are results of environmental considerations and inadequate material resources, (2) company benefits rising from the usage of cheaper supplies, trading of used materials and parts and having a 'green company image'. Apart from the environmental perspective, the common objective of all the recovery processes is to reclaim the economic value as much as possible. So, all the necessary operations (like disassembly, cleaning, sorting, replacing and repairing) for reclaiming the parts and materials need to be performed with minimum possible costs.

A simple classification for recovery proposed by Güngör and Gupta (1999) categorizes recovery into material recovery and product recovery. Those categories are also known as recycling and remanufacturing, respectively. Recycling aims to recover the material content of retired products by performing the necessary disassembly, sorting and chemical operations. On the other hand, remanufacturing preserves the product's (or the part's) identity and performs the required disassembly, sorting, refurbishing and assembly operations to bring the product to a desired level of quality (Güngör and Gupta, 2001a).

Figure 2.1 presents a schematic description proposed by Guide et al. (1999) for representation of the recovery processes.

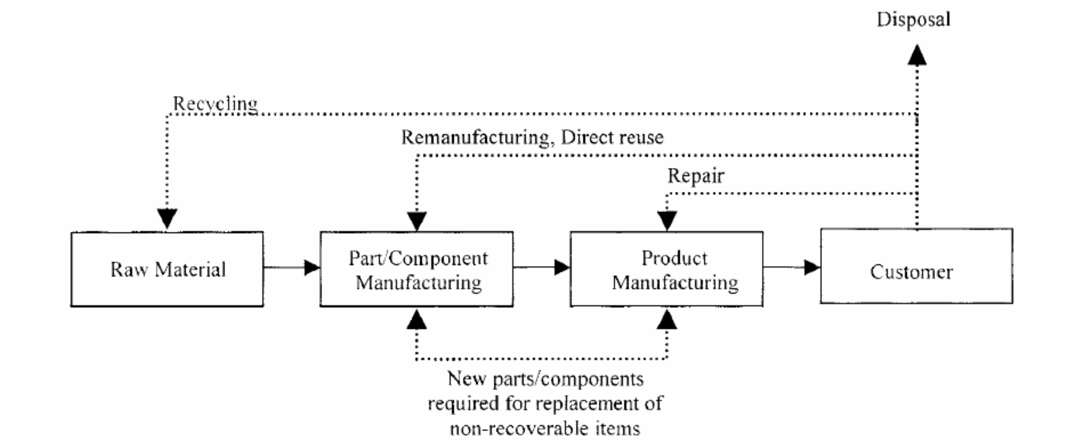


Figure 2.1 Schematic description of recovery processes (excerpted from Guide et al., 1999)

2.2 DISASSEMBLY PROCESS

Among all the product recovery issues the most addressed one is the disassembly of EOL products. This is mainly because disassembly is the first stage in any material recovery process after the product has been returned from the end user (Guide et al., 1999). So all complicated products require some degree of disassembly process regardless of their recovery option. Apart from product recovery, disassembly is also employed in maintenance operations which do not require the object of interest to be an EOL product. Lambert (2006b) defines the three main applications of disassembly studies as: assembly optimization, maintenance and repair, and EOL disassembly.

Güngör and Gupta (1997) define disassembly as a systematic process that allows reusable, non-recyclable, and hazardous subassemblies to be selectively separated from recyclable ones. Güngör and Gupta (1999) define "disassembly as systematic method for separating a product into its constituent parts, components, subassemblies, or other groupings"; more generalizing the process by excluding the reasons behind it. Lambert (2006b) describes disassembly as the non-destructive detachment of components or modules from a product.

De Ron and Penev (1995) simply define the aim of disassembly as to regain the value added to products and materials, and to protect the environment.

Disassembly can be separated into two: (1) Complete Disassembly and (2) Selective or Partial Disassembly. In complete disassembly the product is fully disassembled. However, complete disassembly is not always feasible due to some irreversible assembly operations such as soldiering. In selective disassembly one or more parts and/or subassemblies are removed from a product. Lambert (1999) defines selective disassembly as non destructive, reversible disassembly of complex products into less complex subassemblies or single parts.

Lambert (1997) proposes that in EOL disassembly, within the constraints imposed by legislature, agreements, licenses and the like, the revenues of the total recovery process should be as high as possible.

No disassembly operations are without costs since all disassembly activities require some resources like labor and machinery. The revenue of some components might be low when compared with their corresponding disassembly costs. So recovery problems seek to balance the costs of disassembly with the revenues from material recovered (Navin-Chandra, 1994). Usually, maximum profit is aimed at, subjected to a set of environmental and technical criteria, such as the compulsory recovery of hazardous materials (Lambert, 2006b).

Disassembly processes by their nature bear uncertainty characteristics that need to be taken into account. Those uncertainties include duration of disassembly task times, arrival time, quantity and quality (product and/or component states) of discarded products and estimation of the recovery values of used parts (Lee et al., 2001).

2.3 DISASSEMBLY LEVELING

Complete disassembly may seem to provide the best way of minimizing the damage to the environment. It is not profitable when the cost of disassembly is more than the market and environmental benefits. Thus, it is important to find a balance between the resources invested in a disassembly process and returns realized from it (Gao et al., 2002).

Güngör and Gupta (1999) define disassembly leveling problem as achieving a disassembly level to which the product of interest is disassembled to keep profitability and environmental features of the process at a desired level. In the literature, finding the optimum balance between the resource requirement and the benefit of the disassembly process is studied usually from the cost analysis view point. De Ron and Penev (1995) outline a number of economic considerations for determination of disassembly level as follows:

- the value added to products and materials,
- the disassembly cost per operation,
- the revenue per operation,
- the penalty if poisonous materials are not completely removed.

2.4 DISASSEMBLY SEQUENCING

Disassembly sequencing is critical in minimizing resources (i.e. time and money) invested in disassembly and maximizing the level of automation of the disassembly process and the quality of the parts recovered (Güngör and Gupta, 2001a). Lambert (2003) defines disassembly sequences as the listings of subsequent disassembly actions. Disassembly Sequences are also named as Disassembly Sequence Plans (DSP) or Disassembly Process Plans (DPP) in the literature. In more detail a disassembly sequence is an order of disassembly tasks that begins with a product to be disassembled and terminates in a state where all of the parts of interest are disconnected (thus, it could be either partial or complete disassembly)(Güngör and Gupta, 2001b).

Work on disassembly sequences has been performed for a variety of purposes. Lambert (2003) summarizes those as:

1. remote construction and repair in inaccessible or hazardous environment such as spacecraft and nuclear equipment
2. optimal repair and maintenance
3. a tool for assembly optimization
4. design and optimization of disassembly lines
5. optimum product design regarding the product's end-of-life phase (Design for Disassembly)

Güngör and Gupta (1999) describe the disassembly sequencing problem as to find optimal or near-optimal disassembly sequence, which minimizes the cost of disassembly (assuming that a certain level of disassembly is required) or obtain the best cost/benefit ratio for disassembly.

Moyer and Gupta (1997) state that disassembly sequencing problem is NP-complete when minimum cost is the concern. Disassembly sequencing problem seeks the optimum sequence out of a set of all possible disassembly sequences (represented graphically) via heuristics, metaheuristics, or mathematical programming (Lambert, 2006b).

2.4.1 PRODUCT REPRESENTATIONS

Products should be represented by a certain scheme in order to transform its information into a disassembly process plan. This is called the product representation problem, which is the problem of representing products for disassembly purposes (Lee et al., 2001). Graphs have been a major tool for representing the possible stages of the disassembly process and alternative disassembly strategies for every subassembly (Gao et al., 2002). In the literature a number of methods such as trees, state diagrams, disassembly precedence graphs, AND/OR graphs, process graphs, Petri-nets, disassembly constraint graphs, etc. are used to represent the product structures. The most widely used ones are AND/OR graphs and disassembly precedence graphs.

Homem de Mello and Sanderson (1990) have introduced AND/OR graph to represent all possible assembly and disassembly sequences of a product. In an AND/OR graph there are nodes representing parts or other possible subassemblies of the product and there are edges representing the possible assembly/disassembly operations (see Figure 2.2 for a four part example). An AND/OR graph starts from a complete product and ends with parts completely disassembled. At each subassembly node the out going edges can be in an AND or OR relation.

In disassembly precedence graphs tasks are represented with nodes. An arc is directed from node i to node j if task i is an immediate predecessor of task j (see Figure 2.3). Disassembly precedence graphs are compact, which means that they represent a large number of sequences with a relatively simple graph (Lambert, 2006a).

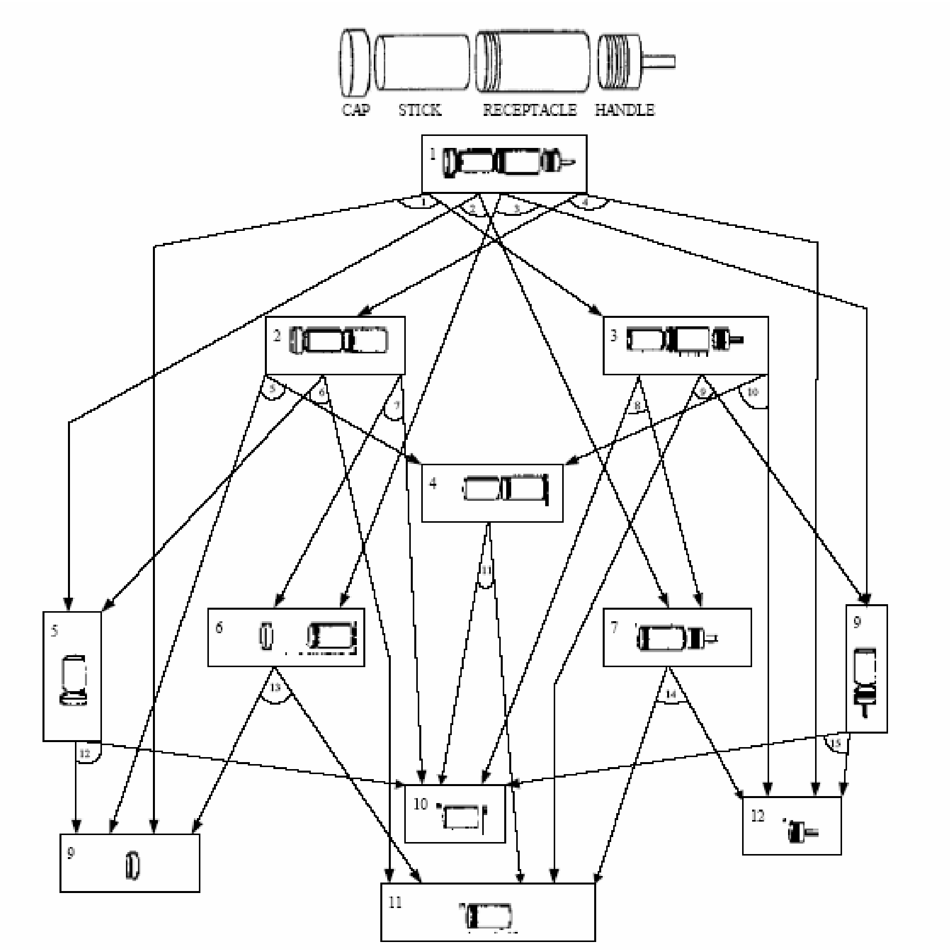


Figure 2.2 Example AND/OR graph (Homem de Mello and Sanderson, 1990)

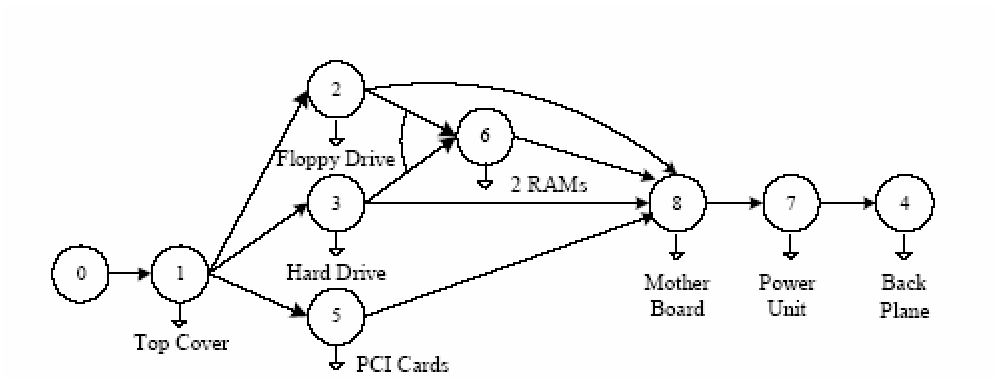


Figure 2.3 Example disassembly precedence graph (Güngör and Gupta, 2002)

2.4.2 REVIEW OF DISASSEMBLY SEQUENCING LITERATURE

Five review papers on the disassembly sequencing problem have been published. The paper by O'Shea et al. (1998) covers mainly the research between years 1993–1996. A second paper, by Güngör and Gupta (1999) summarizes more comprehensively the field of environmentally conscious manufacturing dedicating only part of their work to disassembly sequence planning and focusing between years 1994 and 1998. The third one is by Tang et al. (2000) and discusses papers till year 1998. The fourth study, by Lee et al. (2001) discusses disassembly sequencing along with disassembly scheduling. The fifth survey is due to Lambert (2003) and extensively discusses the studies till year 2002, with an emphasis on the more recent work. In our work, we review previous related work, emphasizing the ones published after year 2001.

Güngör and Gupta (1997) propose a disassembly sequence generation heuristic depending on the precedence relationships of the components and average difficulty level of the removal of the components. They minimize the total time of disassembly due to the changes in the disassembly direction and type of joints broken. Later Güngör and Gupta (2001a) solve the disassembly sequencing problem using a branch-and-bound algorithm. They consider both AND and OR type relations. Their methodology involves the development of the disassembly precedence matrix representing the disassembly constraints by analyzing the product of interest; and generation of near-optimal disassembly sequence plans from disassembly precedence matrices.

Lambert (1997, 1999 and 2002), in his studies, solves the problem optimally. In his first study, he solves the selective disassembly problem using process graphs and a set of profit based decision rules. Later in his work, in 1999, he shows that selective disassembly sequencing problems can be solved optimally using linear programming. In his last work, he presents a method to generate precedence relations from the connection diagrams.

Lambert (2006a and 2006b) concentrates on sequence dependent cost issues in disassembly processes. In his former work he proposes an extension of the exact method (Lambert, 1999) using a binary integer linear programming approach. In his latter study he proposes a two-commodity network flow approach by iteratively solving a binary integer linear programming problem for the sequencing problem.

Lambert and Gupta (2002) study the optimal lot-sizes of EOL products to disassemble so as to fulfill the demand of various components. Their approach is based on disassembly graphs (based on the previous works of Lambert, 1999) finds the optimal number of lot-sizes and disassembly sequence.

Gonzales and Adenzo-Diaz (2005) use bill-of-material based product structure to determine the EOL strategies (depth of disassembly and recovery options). They use a scatter search metaheuristic to obtain a maximum profit disassembly sequence. In their very recent work Gonzalez and Adenso-Diaz (2006), extend the scatter search approach to the optimum disassembly sequence problem of the complex products with sequence-dependent disassembly costs.

Li et al. (2005 and 2006) study disassembly sequence generation for maintenance purposes. In their former work (2005), they propose a disassembly representation scheme, called disassembly constraint graph (DCG) and a genetic algorithm for the near-optimal disassembly sequences of the target component. In their latter work, they propose a reasoning based algorithm which uses the DCG for determining the optimal disassembly sequence.

2.5 DISCUSSION

In this study, we consider only AND type relations thus, no alternative sequence is available for disassembling a part. So our problem reduces to selecting parts for disassembly; which in turn defines the necessary disassembly tasks. Unlike many studies in disassembly literature we assume sequence independent task times.

The selective version of the disassembly sequencing problems determines the optimal sequence and also the set of most profitable parts to recover. There are some selective disassembly sequencing problem applications where minimum cost is aimed but those problems consider other issues like demand or repair/maintenance (see Table 2-1). When time is incorporated into the disassembly sequencing problem, the studies we already reviewed, consider complete disassembly. The studies to date for selective disassembly either consider cost or time, but not both.

In this study we consider time and profit issues together and find the set of parts that maximizes total profit without violating the AND type precedence relations and time constraint. This problem is a special case of the profit oriented selective disassembly line balancing problem, where we have a single workstation, only AND type of precedence relations only and given cycle time.

Table 2-1 Summary of our disassembly sequencing literature reviewed

	Precedence Relations	Disassembly Level	Solution Method	Objective function
Güngör and Gupta (1997)	AND	Complete	Decision based heuristic	Minimum time
Güngör and Gupta (2001a)	AND/OR	Complete	Branch-and-bound	Minimum time
Lambert (1997)	AND/OR	Selective	Graphical	Maximum profit
Lambert (1999)	AND/OR	Selective	Mathematical programming	Maximum profit
Lambert and Gupta (2002)	AND/OR	Selective	Mathematical programming	Minimum cost
Lambert (2006a and 2006b)	AND/OR	Complete	Mathematical programming	Maximum profit
Altekin (2005)	AND/OR	Selective	Heuristic approach	Maximum profit
Gonzales and Adenzo-Diaz (2005 and 2006)	AND/OR	Complete	Scatter search heuristic	Minimum cost
Li et al. (2005)	AND/OR	Selective	Genetic Algorithm	Minimum Number of Operations
Li et al. (2006)	AND/OR	Selective	Decision based heuristic	Minimum Number of Operations

CHAPTER 3

PROBLEM DEFINITION

In this chapter we introduce the terminology, product representations used in our study and define our problems. The mathematical programming formulations together with their underlying assumptions are also provided along with a discussion on the complexity of the problems.

3.1 DISASSEMBLY PROBLEM ENVIRONMENT

3.1.1 TERMINOLOGY

Product is a used or returned item which arrives at a disassembly center for recovery. A component (or a part) is an object which cannot be further broken down into smaller elements. A subassembly of a product is a subset containing two or more parts such that every part is connected with at least one other part with a joint.

The general problem of determining the optimal disassembly tasks and parts is the following: given a product, consisting of a set of components $C = \{C1, C2, \ldots, Cn\}$ matched by a set of joints $J = \{J1, J2, \ldots, Jm\}$ (such as screws, clips, welds, etc.), the aim is to determine which joints must be broken and the components must be removed so as to obtain detached components of interest (Gonzales and Adenso-Diaz, 2006).

A disassembly task can be seen as a decomposition of a product or subassembly into two or more subassemblies and/or parts (Güngör and Gupta, 2001b). For example, removing a joint and freeing subassemblies and/or parts is a disassembly task. The amount of time necessary to complete each disassembly task is the disassembly task time or simply the task time. As disassembly is mostly a labor intensive operation, the task costs are commonly supposed to be proportional to the task times (the proportional factor being the labor rate) (Gonzales and Adenso-Diaz, 2006).

Das et al. (2000) determine seven factors for estimating the cost of disassembly operations. Those factors are: time, tools, fixtures, access, instruction, hazards and force requirements. In their study, Das et al. assign relative importance to each factor, giving the greatest importance to the operation times. As most of the disassembly operations are manual, we consider the disassembly time as a direct measure of the associated labor cost plus direct labor hours are usually used to allocate the overhead costs. Since, we recover materials and parts in a disassembly; we may ignore the direct material cost.

Usually both time and cost values of disassembly operations depend on the preceding operations as well. Sequence dependent costs are frequently encountered in practice, which is intrinsic to the nature of preparatory tasks such as product or gripper reorientation, refixturing and tool exchange typically depend on the configuration that is left by the previous operation (Lambert, 2006a).

3.1.2 PRECEDENCE RELATIONS

A (precedence) relation that is defined among disassembly tasks specifies the sequence of executions. In a feasible disassembly sequence, precedence relationships among the tasks must be satisfied. Lee et al. (2001) state that the precedence relations are mainly due to the geometrical and technological factors in the disassembly process. The relationships that need to be considered in the disassembly case are AND and OR type relations. The representations summarized by Altekin (2005) are as follows.

AND Precedence: Tasks i_1 through i_m are AND predecessors of task i, if all of them must be finished before starting task i (see Figure 3.1).

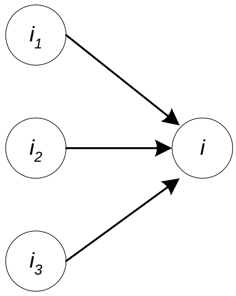


Figure 3.1 An AND precedence relation example

AND Succession: Tasks i_1 through i_m are AND successors of task i, if all of them can be performed, regardless of each other, after finishing task i (see Figure 3.2).

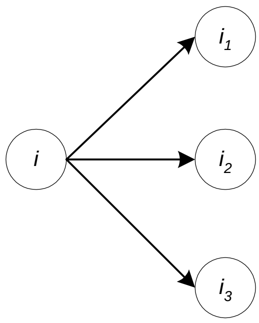


Figure 3.2 An AND succession relation example

OR Precedence: Tasks i_1 through i_m are OR predecessors of task i, if at least one of them must be finished before starting task i (see Figure 3.3).

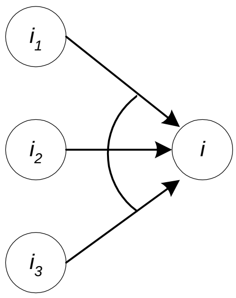


Figure 3.3 An OR precedence relation example

OR Succession: Tasks i_1 through i_m are OR successors of task i, if at most one of them can start after task *i* is finished (see Figure 3.4).

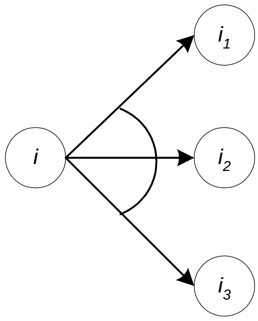


Figure 3.4 An OR succession relation example

There may exist various combinations of the relations in a disassembly process. These relations are called as complex AND/OR relationships. A complex AND/OR relationship exists between task i_1; task i_2 and task i_3, in relation to task *i*, if task i_1 along with either task i_2 or task i_3 must be completed prior to task i_4 as illustrated in Figure 3.5.

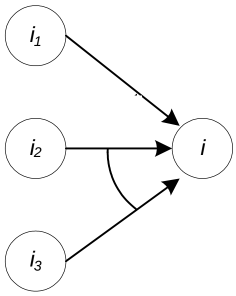


Figure 3.5 A Complex AND/OR relation example

Precedence relationships also exist in the traditional assembly line balancing problem (ALBP). Even though approaching disassembly as the reverse of assembly may sound reasonable, the operational characteristics of the disassembly and assembly are quite different (Güngör and Gupta, 1999) for complex products. The precedence relationships in the traditional ALBP are usually limited to simple AND types. These relationships are developed considering the physical and functional constraints, because the objective of an assembly process is to create a stable (and functional) end product. However, functionality is not important in disassembly processes so the only constraints considered are geometrical and technological factors.

In the ALBP, AND relationships among parts are usually represented by an acyclic graph in which nodes represent assembly tasks and unidirected edges connecting two nodes represent the precedence relationship between the two tasks. A binary matrix form of the precedence graph is also used to express these relationships (Güngör and Gupta, 2001b). In this study, we also focus on only AND relations.

3.2 DISASSEMBLY PART SELECTION PROBLEM

Disassembly part selection is the selection of the parts and the associated set of the tasks in order to have the maximum profit (revenue due to parts and cost due to tasks). We study two versions of the disassembly part selection problem. The first version is the simple selection problem in which we are interested in maximizing the profit obtained by disassembling a product without any constraints except the AND precedence relations. In the second problem, we introduce a time constraint and try to determine the most profitable set of disassembly tasks within the given time.

We have introduced a cycle time constraint in our problem to emphasize the relation between time and profit. For infinite demand case, cycle time constraint concerns with the profit/time ration for each product. When demand is finite and various parts/materials obtained from a product, cycle time can be used with the associated disassembly rate to satisfy the demand as much as possible. Moreover, cycle time is important where the values of some recovered products decrease over time.

In this problem we assume a single workstation (disassembly cell) for disassembling the product within given cycle time. This assumption applies to the majority of today's disassembly systems consist of a single workstation (Das and Caudill, 1999). Single workstations provide highly flexible disassembly environments, yet they lack the high productivity rates that can be satisfied in the practice by disassembly lines.

Since representing disassembly precedence relation practically is an issue. In our problem, we consider only AND type precedence relations between tasks for simplicity. For performing a disassembly part selection problem one must determine all possible disassembly processes for a product so every product need to be analyzed for disassembly. However assembly precedence graphs are readily available and they can be reversed for disassembly purposes. Since classical assembly precedence graphs are only comprised of AND type precedence relations and when reversed, the disassembly graphs will be also comprised of AND type relations. But this approach may not apply as some assembly operations are irreversible (like soldering etc.). Yet those kinds of operations can be deleted from the assembly diagrams for practical use. However, we should note that there are also other disadvantages of considering only AND type precedence relation. For assembly systems alongside the physical criteria, there are also some functionality constraints but in disassembly the functionality constraint is relaxed so OR type relations are commonly encountered. The exclusion of OR type relations may yield costly recovery of parts, which is addressed as a further research issue.

Our problem is concerned with the complete discarded product to be disassembled. This EOL product is composed of m parts, all of which has positive part revenues. Disassembly process is comprised of a total of n disassembly tasks necessary to disassemble the product completely down to the part level. Not all disassembly tasks release parts and a task can release more than one part when realized.

The disassembly tasks will be characterized by their task costs and task times. The released parts will have revenues that can be obtained when all the necessary tasks are performed. The disassembly times and costs of each task, the precedence relationships among the tasks and the revenues of each part are known. All parts have positive revenues but not all the parts are clearly profitable to be recovered, due to their associated disassembly costs. Depending on the revenue and cost values, the disassembly having the maximum profit can be partial. Cycle time, the amount of time available to complete the chosen set of disassembly tasks, is given for the problem instances.

To represent our problem we will use the disassembly precedence graphs (see Figure 3.6) in which both the tasks and the parts are represented as nodes. White nodes (i = 1,2,…,n) are for the disassembly tasks and the grey nodes (j = 1,2,…,m) for the parts.

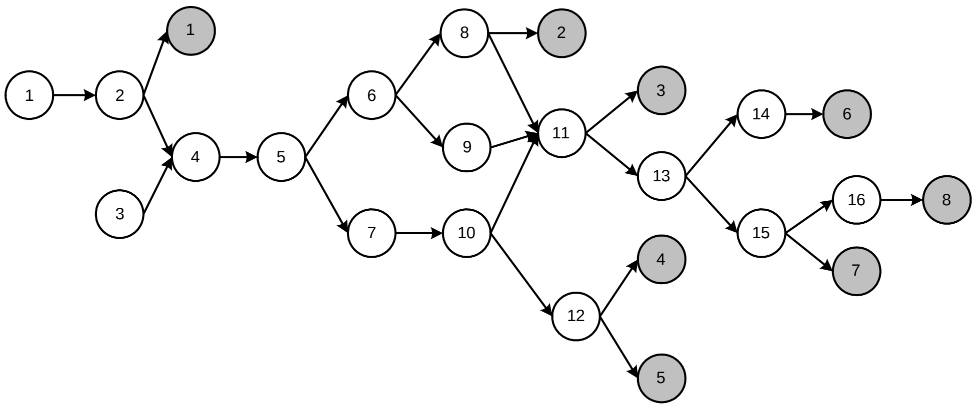


Figure 3.6 A sample disassembly precedence graph

The arc leaving a task node (i.e., task 1) and entering another task node (i.e. task 2) represents the precedence relationship between the corresponding nodes (i.e. task 1 must be completed in order to perform task 2). The arc leaving a task node and entering a part node represents the immediate part release with the realization of tasks (i.e., when task 2 is completed part 1 is released).

The main assumptions are summarized as follows:

A1. A single discarded product will be disassembled.

A2. There is infinite demand for each recovered part.

A3. All the disassembly will be performed in a single workstation.

A4. The cycle time is given.

A5. The disassembly precedence relations are known and there is only AND type relationships.

A6. The part revenues, sequence independent task times and costs are deterministic and known.

A7. Each part has a positive revenue and there is no disposal cost for the parts and/or subassemblies that will not be salvaged.

A8. Not all disassembly tasks release parts and a task can release more than one part as well as two different tasks can release the same part.

3.3 PROBLEM I: NO TIME CONSTRAINT

3.3.1 NOTATION AND FORMULATION

Following is the related notation

Indices:

j	Part index	$j = 1,2,\ldots,m$
i	Task index	$i = 1,2,\ldots,n$

Parameters:

p_j	Revenue of part j
c_i	Cost of disassembly task i
S_j	Set of tasks that needs to be performed in order to recover part j

Decision Variables:

$$x_j = \begin{cases} 1, & \text{if part } j \text{ is recovered} \\ 0, & \text{otherwise} \end{cases}$$

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

The mathematical programming formulation of the first problem is given below:

$$\text{Max} \quad \sum_{j=1}^{m} p_j\, x_j - \sum_{i=1}^{n} c_i y_i \tag{3.1}$$

$$\text{s.t.} \quad \sum_{i \in S_j} y_i - \left[\left| S_j \right| - 1 \right] - x_j \geq 0 \quad \forall\, j\,,\ i \in S_j \tag{3.2}$$

$$x_j \in \{0,1\} \quad \forall\, j \tag{3.3}$$

$$y_i \in \{0,1\} \quad \forall\, i \tag{3.4}$$

This integer programming formulation is comprised of two major equations. The first one is the objective function equation, equation (3.1) , which is the sum of the recovered part revenues minus the cost of realized tasks. The second equation, equation(3.5), is the constraint which implies that a part is to be released if and only if all of its associated tasks are performed. In this formulation there is a single inequality written for each part, so there are m constraints.

In order to obtain a totally unimodular constraint matrix, we will replace each equation of (3.2) with a new constraint set. Our new constraint set will have an inequality for each part and task association, stating that a part can only be recovered only if each associated task is realized. Our new formulation is as follows.

$$\text{Max} \quad \sum_{j=1}^{m} p_j\, x_j - \sum_{i=1}^{n} c_i y_i \tag{3.1}$$

$$\text{s.t.} \quad y_i - x_j \geq 0 \quad \forall\, j\,,\ i \in S_j \tag{3.5}$$

$$x_j \in \{0,1\} \quad \forall\, j \tag{3.3}$$

$$y_i \in \{0,1\} \quad \forall\, i \tag{3.4}$$

This formulation solves for m+n binary variables. The number of constraints in our formulation is determined by the number of precedence relations in the problem. In the worst case, there will be $(m \cdot n)$ constraints in (3.5).

3.3.2 TOTAL UNIMODULARITY

In our formulations, constraints (3.2) and (3.5) force y_i's to be greater than or equal to associated x_j's, while the objective function is forcing y_i's to be as small as possible. As a result y_i's will be equal to associated x_j's, always being assigned a binary value due to constraint (3.3) . Therefore, constraint set (3.4) is redundant, for both formulations.

As explained earlier, constraint (3.5) corresponds to the relation between a part and its associated tasks. This relationship can be expressed graphically (in Figure 3.7) as a bipartite graph. The transpose of an incidence matrix A^T of a bipartite graph is composed of rows representing arcs and columns representing vertices, where $A^T_{a,v}$=1(-1) if and only if a enters (leaves v) and zero otherwise. With this definition on mind, constraint (3.5) can be reformulated as below:

$$A^T \begin{bmatrix} y \\ x \end{bmatrix} \geq 0 \qquad (3.6)$$

The incidence matrix of a bipartite graph (hence its transpose) is totally unimodular yielding an integral optimal solution if LP relaxation is to be solved (Schrijver, 1998). So the integrality constraints on variables x_j and y_i can be dropped.

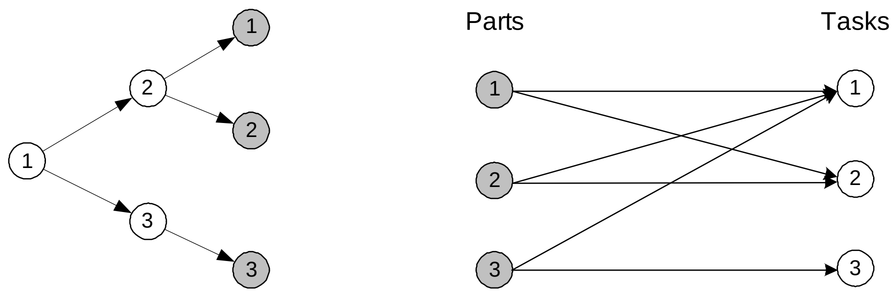


Figure 3.7 Graph representation of the relation between parts and tasks

The reformulation of our first disassembly part selection problem is as follows.

$$\text{Max} \quad \sum_{j=1}^{m} p_j \, x_j - \sum_{i=1}^{n} c_i y_i \tag{3.1}$$

$$\text{s.t.} \quad y_i - x_j \geq 0 \quad \forall\, j,\ i \in S_j \tag{3.5}$$

$$x_j \leq 1 \qquad \forall\, j \tag{3.7}$$

$$x_j \geq 0 \qquad \forall\, j \tag{3.8}$$

$$y_i \text{ urs} \qquad \forall\, i \tag{3.9}$$

Note that the coefficient matrix (for (3.5) and (3.7)) of the above formulation is $\left[\dfrac{A^T}{\underline{0} \mid I}\right]$, which is still totally unimodular.

Like our mixed integer problem formulation, this formulation also solves for *m*+*n* variables. In this formulation, however, the variables are continuous. The relaxation does not change the constraint set (3.5); thus there are still $(m \cdot n)$ constraints for the worst case.

Lambert (1999) treats a similar problem with subassemblies as profit generating objects aside with the end parts. Lambert also defines OR relations between activities. His objective function is the same: the sequence of actions that generates maximum net revenue. Lambert states that his formulation can be solved for integral solutions by using the relaxation of the 0-1 constraints. However, Lambert does not realize that the associated coefficient matrix (network matrix) is totally unimodular.

3.3.3 DUAL

In Section 3.3.2, we proposed a linear programming formulation to our problem by using the total unimodularity characteristic of the constraint matrices. Here, we will mechanically formulate and then describe the dual formulation for the problem.

Using the duality theory, we can find the following formulation by employing our dual variables α'_{ji} and β_j for constraint sets (3.5) and (3.7) respectively.

$$\text{Min} \quad z = \sum_{j=1}^{m} \beta_j + \sum_{j=1}^{m} \sum_{i \in S_j} 0.\alpha'_{ji} \tag{3.10}$$

$$\text{s.t.} \quad \beta_j - \sum_{i \in S_j}^{n} \alpha'_{ji} \geq p_j \qquad \forall j \tag{3.11}$$

$$\sum_{j \in U_i} \alpha'_{ji} = -c_i \qquad \forall i \tag{3.12}$$

$$\beta_j \geq 0 \qquad \forall j \tag{3.13}$$

$$\alpha'_{ij} \leq 0 \qquad \forall j, i \tag{3.14}$$

Here U_i is the set of parts that need task *i* to be performed for their recovery.

Let $\alpha_{ij} = \dfrac{-\alpha'_{ij}}{c_i}$, then we have:

$$\text{Min} \quad z = \sum_{j=1}^{m} \beta_j \tag{3.10}$$

s.t. $\beta_j - p_j + \sum_{i \in S_j}^{n} \alpha_{ji} c_i \geq 0 \qquad \forall j \qquad (3.10')$

$$\sum_{j \in U_i} \alpha_{ji} = 1 \qquad \forall i \qquad (3.11')$$

$$\beta_j \geq 0 \qquad \forall j \qquad (3.13)$$

$$\alpha_{ij} \geq 0 \qquad \forall j, i \qquad (3.13')$$

In the formulation, α_{ji}, represents the percentage of the cost of task i that is compensated by part j and β_j represents the positive profit margin of part j. The significance of these variables can be better understood by the following discussion.

As we are dealing with selective disassembly, in the optimal solution we will have some parts recovered with their associated realized tasks and other parts not recovered. In the optimal solution recovery decisions of a part can be defined as follows:

- A part (part j) can be recovered on its own as long as its revenue is higher than the total task costs required for its recovery. Also the parts that require no other additional task to be realized (more than that already performed for the recovery of part j) will also be recovered alongside part j.
- A set of parts can cover their recovery expenses altogether (in this case, each part can cover a portion of a task necessary for their recovery), so it will not be profitable to produce any of the parts in this set on its own.

Depending on those recovery decisions we can say that no matter how a part is recovered, task costs are shared between parts. So we can write a profit margin equation for each part as follows:

$$\hat{L}_j = p_j - \sum_{i \in S_j}^{n} \alpha_{ji} c_i \qquad (3.15)$$

Here α_{ji} represents the percentage of the task i that is compensated by part j ($0 \le \alpha_{ji} \le 1$). A part is economically profitable if and only if the profit margin $\hat{L}_j$ is positive or zero. In the optimal solution, $\hat{L}_j < 0$ means that part j is not going to be recovered along with the parts needing the tasks represented in the profit margin equation of part j. Therefore we can identify two part groups in any optimal solution: parts recovered $\hat{L}_j \ge 0$ and not recovered $\hat{L}_j \le 0$. Thus, the optimal profit obtained can also be represented as below:

$$z = \sum_j \max(\hat{L}_j, 0), \qquad (3.9')$$

where $\max(\hat{L}_j, 0)$ is the positive profit margin for part j and so is our dual variable β_j which can also be defined as the amount of profit contribution of part j to the overall profit.

To sum up, if we can find the optimal share (by using the dual formulation) we can identify the optimal set of tasks and parts. In the dual problem formulation, minimizing the objective function value forces the tasks to be assigned to the parts having positive profit margins as long as possible. In the optimal share, all the tasks that will be performed in the optimal solution are covered 100% by the parts having positive profit margins. The optimal dual objective function value is the exact profit value for the primal problem. However, for the decision of parts to be produced, one more step needs to be performed: If $\beta_j \ge 0$ and the slack variable for equation (3.10') is zero, then $x_j = 1$, otherwise $x_j = 0$.

To better understand the dual approach, an analogy by using a bin packing problem is helpful: we have bins and objects (different types of grains) that need to be packed for shipping. In our problem, specific type of grains can only be placed in specific type of bins. A type of grain can be shipped only if the entire associated amount is packed inside the bins and so no percentage of the grain is left outside. Bins are standing in the place of parts in this problem and the reason we are trying to maximize them is that each part, being recovered brings in profit. We maximize the number of bins to be shipped, which is equivalent to that we have to minimize the amount of grains left. Altogether, what we need to minimize is the empty space left in the bins while the empty space is analogous to the profit margin in our problem.

3.4 PROBLEM II: CYCLE TIME CONSTRAINT

When a time constraint (cycle time (*CT*)) is introduced, our formulation's coefficient matrix, is unfortunately, no longer totally unimodular.

We need to define the following additional parameters:

CT	Cycle time
t_i	Time of disassembly task i

Our mathematical formulation is as expressed below:

$$\text{Max} \quad \sum_{j=1}^{m} p_j\, x_j - \sum_{i=1}^{n} c_i y_i \tag{3.1}$$

$$\text{s.t.} \quad y_i - x_j \geq 0 \qquad \forall\, j,\; i \in S_j \tag{3.5}$$

$$\sum_{i=1}^{n} t_i y_i \leq CT \tag{3.16}$$

$$x_j \in \{0,1\} \qquad \forall\, j \tag{3.3}$$

This formulation solves for *m*+*n* variables. In the worst case, there are $(m \cdot n) + n$ constraints.

With the addition of the time constraint (3.16), we can approach our problem from a different view point. We have a time window for which we need to find a set of disassembly tasks to be performed in order to have the maximum profit. As parts are natural consequences of the disassembly tasks, we can rephrase our aim as finding the most profitable parts that can be recovered within the given cycle time.

We will approach this problem as a modified knapsack problem where parts are the objects that need to be placed inside the knapsack and the associated task times represent the volumes of the objects. The benefit that can be obtained from an object is the revenue of this part minus the costs of the necessary tasks. Unlike the conventional knapsack problem, volumes and benefits of the objects depend on the other objects in our case.

As knapsack problems are *NP*-complete (Dutta and Majumder, 1998), there is no easy way for solving them in polynomial time. One has to rely on implicit enumeration techniques like branch and bound algorithms and dynamic programming approaches. We propose a branch and bound algorithm for finding the optimal selection of parts together with their tasks in this study.

CHAPTER 4

SIZE REDUCTION MECHANISIMS

Disassembly part selection is the act of selecting the set of parts and the necessary tasks to be performed, for disassembling a product so as to maximize the profit. In this chapter we introduce some mechanisms for reducing the size of the problem instances.

4.1 REDUCING DISASSEMBLY GRAPHS

When disassembly precedence graphs are examined for simplification purposes, there are possible ways to reduce the size of the graph by eliminating and merging both tasks and parts.

A similar idea is also employed by Lambert (1997). He optimally solves the disassembly sequencing problem on process graphs. His graphical method is partly comprised of inserting the activity costs with in part revenues and processing the graph starting from end nodes (parts) towards the activity nodes (tasks). Lambert's idea of adding the activity costs to part revenues can be used to reduce the number of tasks. The reduction procedure can be extended beyond this simple idea by concentrating on the complete disassembly graph rather than processing it from parts towards tasks. We present some reduction rules after we define additional terminology.

Candidate Tasks Set (*T*): Set of tasks on the graph whose execution decisions are not-yet-made.

Candidate Parts Set (*P*): Set of parts on the graph whose recovery decisions are not-yet-made.

DO Set (P^*): Set of profitable parts.

DON'T Set (P'): Set of parts which are not profitable.

Performed Tasks (T^*): Set of tasks that for the recovery of parts in the DO Set.

Necessary Tasks Set (S_j): Set of tasks necessary for the recovery of part *j*

Basic Profit (L_j): The summation of part revenues minus the costs of tasks necessary for the recovery of that specific part ($L_j = p_j - \sum_{i \in S_j} c_i$; where p_j is revenue of part *j* and c_i is cost of task *i*).

Graph Reduction Rules:

1. **Direct Recovery Control:**

   If a part has a net profit greater than or equal to zero ($L_j \geq 0$) then that part can compensate all task costs in its respective Necessary Tasks Set. In other words, if this part is already profitable without any contribution from other candidate parts then it is selected for retrieval and can be removed from the Candidate Parts Set and added to the DO Set. Thereafter all tasks in the Necessary Task Set of part *j* removed from the Candidate Task Set and added to the set of tasks to be performed. This reduction rule can only be employed in Problem I.

2. **Part Time Control:**

   The parts whose required tasks have a total of more than *CT* units cannot be recovered, and are added to the DON'T Set. This reduction rule can be employed in Problem II only.

3. **Task and Part Reduction:**

   If task *i* has a cost greater than the sum of the revenues of the parts that require task *i* for recovery, there is no profitable sequence that can justify the cost of task *i*. In such a case, all parts requiring task *i* and task *i* itself should be discarded from the Candidate Parts Set and the Candidate Tasks Set, respectively. The discarded parts must be added to the DON’T Set.

4. **Task Insertion into Parts:**

If task *t* is only needed for the recovery of a specific part *j* then cost of task *t* is deduced from the revenue of part *j* (revenue *j* = revenue *j* – cost *t*) and task *t* is discarded from the Candidate Tasks Set (see Figure 4.1, where task *t* is inserted into part *j* and task *t* is discarded). Also the task time will be assigned to part *j* as a part time or will be added to the part time if part *j* is already merged with another task. If this action results with negative revenue for a part then the part is removed from the Candidate Parts Set. (This action reduces to the *Task and Part Reduction* when only one part can be recovered by performing task *i*.)

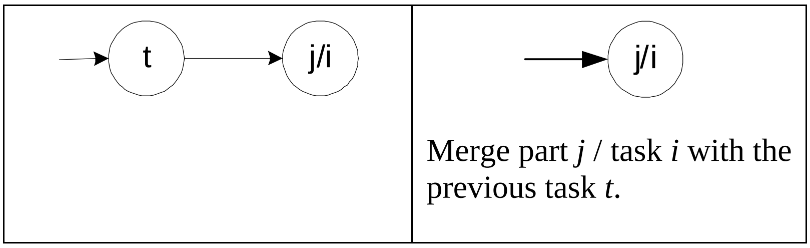


Figure 4.1 Graphical application of task aggregation and task insertion into parts

5. **Part Aggregation:**

If there are parts having the same set of tasks, then they can be aggregated. As one of the parts is kept the remaining parts are discarded from the Candidate Parts Set (see Figure 4.2, where parts *k* and *l* are aggregated into part *k* and part *l* is discarded). The kept part will have a revenue equal to the sum of the merged parts’ revenues. When parts are merged, if available, their part times are also summed. This part of the rule only applies when a part is previously merged with a task and gained a part time value.

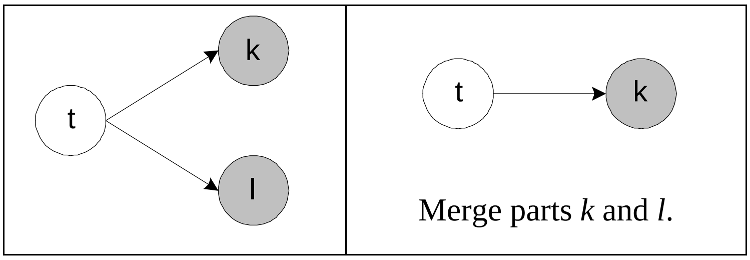


Figure 4.2 Graphical application of part aggregation

6. **Task Aggregation:**

   If there are tasks that have to be performed sequentially in the disassembly graph (with no other tasks in between) and if they are required by the same parts, then they could be merged into one task having a cost and time equal to the sum of the merged tasks' costs and times. The remaining task should be discarded from the Candidate Tasks Set (see Figure 4.1, where task *j* and *t* are aggregated and task *t* is discarded).

4.2 REDUCTION ALGORITHM

In this section, we present an algorithm for implementing the reduction rules.

4.2.1 NOTATION

D_j	Dummy set for task control
MPP_j	Parts that are merged with part *j*
MPT_j	Tasks that are merged with part *j*
MTT_i	Tasks that are merged with task *i*
P	Set of all parts
P^*	Set of parts to be recovered in the solution
P'	Set of parts not to be recovered in the solution
$PAND(i)$	Set of AND predecessors of task *i*
T	Set of all tasks

T^*	Set of tasks to be performed in the solution
S_j	Set of tasks that needs to be performed in order to recover part j
SAND(i)	Set of AND successors of task i
tp_j	Time of part j
U_i	Set of parts that needs task i to be performed in order to be produced

4.2.2 REDUCTION ALGORITHM

Step 1. **// Direct Recovery Control //**

Calculate $L_j \ \forall \ j \in P$

For all $j \in P$

If $L_j \geq 0$ then

$$P = P \setminus \{j\}$$

$$P^* = P^* \cup \{j\}$$

$$T^* = T \cup S_j$$

For all $j \in P \quad S_j = S_j \setminus (S_j \cap T^*)$

Step 2. **// Part Time Control //**

If $\sum_{i \in S_j} t_i + tp_j > CT$ then

$$P = P \setminus \{j\}$$

Step 3. **// Task and Part Reduction //**

For all $i \in T$:

If $c_i > \sum_{j \in U_i} p_j$ then

$$P = P \setminus U_i$$

$$P' = P' \cup U_i$$

$$T = T \setminus \{i\}$$

Step 4. **// Task Insertion Into Parts //**

For all $j \in P$:

$$D_j = S_j \setminus \left[S_j \cap (\bigcup_{l \neq j} S_l) \right] \quad \forall\, j \in P$$

$$S_j = S_j \setminus D_j \quad \forall\, j \in P$$

$$MPT_j = MPT_j \cup D_j$$

$$p_j = p_j - \sum_{i \in D_j} c_i \quad \forall\, j \in P$$

$$tp_j = tp_j + \sum_{i \in D_j} t_i$$

If $p_j \leq 0$ then $P = P \setminus \{j\}$

Step 5. **// Part Aggregation //**

For all $j \in P$, for all $l \in P$

If $S_j - S_l = \emptyset$ and $tp_j = 0$ and $tp_l = 0$ then

$$p_j = p_j + p_l$$

$$MPP_j = MPP_j \cup \{l\}$$

$$S_l = \emptyset$$

$$P = P \setminus \{l\}$$

Step 6. **// Task Aggregation //**

For all $i, k \in T$:

If $U_i = U_k$ and $SAND(i) = \{k\}$ and $PAND(k) = \{i\}$ then

$$c_i = c_i + c_k$$

$$t_i = t_i + t_k$$

$$SAND(i) = SAND(k)$$

$$MTT_i = MTT_i \cup \{k\}$$

For all $l \in SAND(k)$

$$PAND(l) = [PAND(l) \setminus \{k\}] \cup \{i\}$$

For all $j \in U_k$

$$S_j = S_j \setminus \{k\}$$

$$T = T \setminus \{k\}$$

If no update in sets P and T since **Step 1 in current iteration loop,** then **STOP**

Else go back to **Step 1.**

The Reduction algorithm is illustrated on an example given in Appendix A.

CHAPTER 5

AN APPROACH TO TIME CONSTRAINED PART SELECTION PROBLEM

The addition of a time constraint into the simple disassembly part selection problem results in a modified knapsack problem where part and task selection are dependent. In this chapter, we present a branch-and-bound algorithm for solving the disassembly part selection problem with time constraint.

5.1 BRANCH-AND-BOUND ALGORITHM

In our branch-and-bound (B&B) algorithm, the disassembly precedence matrix (DPM) for tasks and part releasing tasks matrix (PN) are used as inputs. Moreover we have part revenues, task costs and task times and finally the cycle time.

The precedence matrix used is similar to the one used in ALBP. We define our matrix by immediate successors set. DPM is mathematically represented by R=[r_{ij}], where the size of the square matrix is $n \times n$, the number of tasks.

$$r_{ij} = \begin{cases} 1, & \text{if task } i \text{ precedes task } j \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

DPM and PN are used for initial problem reduction purposes as well as formation of another matrix (PT), which identifies all the tasks necessary for the recovery of a part, PT=[pt_{ji}] j=1,…,m i=1,…,n where m is the number of parts and n is the number of tasks. As there are only AND type relations in the problem, DPM can directly be transformed into PT.

$$pt_{ji} = \begin{cases} 1, & \text{if task } i \text{ must be performed for the recovery of part } j \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

The B&B algorithm starts from the root node where none of the parts are yet selected for disassembly. We follow a variable dichotomy in the branching process. From each node two children are created. One child represents the case where the selected part (the part that is branched on) is recovered ($x_j = 1$) and the other child represents the case in which the part is not recovered ($x_j = 0$). Branching decision is based on the selection of the part having the greatest profit/time ratio represented below:

$$\max\left\{ \frac{p_j - \sum_{i \in S_j} \frac{c_i}{|U_i|}}{\sum_{i \in S_j} t_i},\ 0 \right\} \tag{5.3}$$

Each node in the search tree is characterized by three sets which represent the parts that are going to be recovered, parts that will never be recovered and the parts waiting for their recovery decision. For each sub-problem not yet searched, those sets need to be stored. For large problems, this requires extensive memory so depth-first-search (dfs) strategy is employed in order to minimize the memory requirements.

The problem reduction mechanisms introduced in Chapter 4 are employed. We initially make use of the reduction rules to cut down the size of the tree before starting B&B. This reduces the size of the search tree, directly lowering the number of parts by means of aggregation rules and deleting the non-profitable and infeasible parts (due to cycle time constraint). We also employ reduction mechanisms for each created sub-problem by taking care of the other parts whose recovery decisions are already settled. All the sets employed in the reduction algorithm (S_j, U_i, $PAND(i)$, $SAND(i)$) can be generated using DPM and PN.

We employ the following part collection rule after selecting the '$x_j = 1$' branch. As we make part recovery decisions, we automatically make task realization decisions associated with the tasks required by part j. That is, once a part j is chosen for recovery it means all the tasks necessary for its recovery ($i \in S_j$) are performed. If there is a part, say part k, having a task requirement which is subset of another ones, say part j (i.e., $S_k \subset S_j$), the recovery of part j covering all tasks necessary for the recovery of part k. Hence, part k can be directly recovered and added to DO Set.

For each sub-problem that cannot be eliminated by reduction mechanisms, we calculate an upper bound on profit. The upper bounding procedures are explained in the following subsection.

5.1.1 UPPER BOUNDS

We employ three upper bounding schemes for our problem. The first two upper bounds relax the cycle time constraint.

Upper Bound 1:

Upper Bound 1 (UB1) evenly partitions the task costs between the associated parts assuming that each part takes the same share. The maximization operator is needed because we are dealing with the partial disassembly and with this apportionment all the parts may not be profitable. We express UB1 as follows:

$$UB1 = \sum_{j \in P} \max\left\{ p_j - \sum_{i \in S_j} \frac{c_i}{|U_i|} \ , \ 0 \right\} \tag{5.4}$$

Theorem 1. UB1 is a valid upper bound on the optimal objective function value, z^*.

Proof.

$UB1 = \sum_{j \in P} \max\{\varepsilon_j \ , \ 0\}$, where $\varepsilon_j = p_j - \sum_{i \in S_j} \frac{c_i}{|U_i|} \quad ; \forall j \in P$.

The objective function value for the optimal solution can be expressed as:

$z^* = \sum_{j \in P^*} p_j - \sum_{i \in T^*} c_i$, where P^* is the set of parts selected in the optimal solution and T^* is the set of tasks required to recover the parts in P^*, i.e. $T^* = \bigcup_{j \in P^*} S_j$

Alternatively objective function can be expressed as:

$$z^* = \sum_{j \in P^*} \left(p_j - \sum_{i \in S_j} \frac{c_i}{|U_i|} + \sum_{i \in S_j} \frac{c_i}{|U_i|} \right) - \sum_{i \in T^*} c_i$$

$$z^* = \sum_{j \in P^*} \left(p_j - \sum_{i \in S_j} \frac{c_i}{|U_i|} \right) + \sum_{j \in P^*} \left(\sum_{i \in S_j} \frac{c_i}{|U_i|} \right) - \sum_{i \in T^*} c_i$$

$$z^* = \sum_{j \in P^*} \varepsilon_j - \sum_{i \in T^*} c_i + \sum_{j \in P^*} \left(\sum_{i \in S_j} \frac{c_i}{|U_i|} \right)$$

Note that:

$\sum_{i \in T^*} c_i = \sum_{i \in P^*} \left[\sum_{i \in S_j} \frac{c_i}{|V_i|} \right]$, where V_i is the set of parts in P^* that require task *i.*

$$\sum_{i \in P^*} \left[\sum_{i \in S_j} \frac{c_i}{|V_i|} \right] \geq \sum_{j \in P^*} \left(\sum_{i \in S_j} \frac{c_i}{|U_i|} \right) \quad \text{as} \quad |U_i| \geq |V_i|$$

This follows;

$$\sum_{i \in T^*} c_i \geq \sum_{j \in P^*} \left(\sum_{i \in S_j} \frac{c_i}{|U_i|} \right)$$

If we define a new variable ε^-

$$\varepsilon^{-} = \sum_{i \in T^*} c_i - \sum_{j \in P^*} \left(\sum_{i \in S_j} \frac{c_i}{|U_i|} \right); \ \varepsilon^{-} \geq 0$$

The objective function value of the optimal solution can be expressed as:

$$z^* = \sum_{j \in P^*} \varepsilon_j - \varepsilon^{-}$$

The fact that $UB1 \geq z^*$ can be seen easily from following.

As $P^* \subseteq P$, where P is the set of all parts,

$$\sum_{j \in P} \max\{\varepsilon_j \ , \ 0\} \geq \sum_{j \in P^*} \max\{\varepsilon_j \ , \ 0\} \geq \sum_{j \in P^*} \varepsilon_j \geq \sum_{j \in P^*} \varepsilon_j - \varepsilon^{-} \quad \therefore \ UB1 \geq z^* \qquad \blacksquare$$

Upper Bound 2:

Upper Bound 2 (UB2) assumes that part revenues are shared equally between all their associated tasks.

$$UB2 = \sum_{i \in T} \max\left\{ \sum_{j \in U_i} \frac{p_j}{|S_j|} - c_i, \ 0 \right\} \tag{5.5}$$

Theorem 2. UB2 is a valid upper bound on the optimal objective function value, z^*.

Proof.

Note that:

$UB2 = \sum_{i \in T} \max\{\varepsilon_i \ , \ 0\}$, where $\varepsilon_i = \sum_{j \in U_i} \frac{p_j}{|S_j|} - c_i \ ; \forall i \in T$.

Objective function value for the optimal solution can be expressed as

$$z^* = \sum_{j \in P^*} p_j - \sum_{i \in T^*} c_i$$

Objective function can also be expressed as:

$$z^* = \sum_{j\in P^*} p_j + \sum_{i\in T^*}\left(\sum_{j\in U_i}\frac{p_j}{|S_j|} - \sum_{j\in U_i}\frac{p_j}{|S_j|} - c_i \right)$$

$$z^* = \sum_{j\in P^*} p_j - \sum_{i\in T^*}\left(\sum_{j\in U_i}\frac{p_j}{|S_j|} \right) + \sum_{i\in T^*}\left(\sum_{j\in U_i}\frac{p_j}{|S_j|} - c_i \right)$$

$$z^* = \sum_{i\in T^*}\varepsilon_i + \sum_{j\in P^*} p_j - \sum_{i\in T^*}\left(\sum_{j\in U_i}\frac{p_j}{|S_j|} \right)$$

Note that:

$$\sum_{j\in P^*} p_j = \sum_{i\in T^*}\left(\sum_{j\in V_i}\frac{p_j}{|S_j|} \right)$$

$$\sum_{i\in T^*}\left(\sum_{j\in U_i}\frac{p_j}{|S_j|} \right) \ge \sum_{i\in T^*}\left(\sum_{j\in V_i}\frac{p_j}{|S_j|} \right), \text{ as } V_i \subseteq U_i$$

This follows;

$$\sum_{i\in T^*}\left(\sum_{j\in U_i}\frac{p_j}{|S_j|} \right) \ge \sum_{j\in P^*} p_j$$

If we define a new variable ε^-

$$\varepsilon^- = \sum_{i\in T^*}\left(\sum_{j\in U_i}\frac{p_j}{|S_j|} \right) - \sum_{j\in P^*} p_j \text{ ; } \varepsilon^- \ge 0$$

The objective function value of the optimal solution can be expressed as:

$$z^* = \sum_{i\in T^*}\varepsilon_i - \varepsilon^-$$

The fact that $UB2 \ge z^*$ can be seen easily from following.

As $T^* \subseteq T$, where T is the set of all tasks,

$$\sum_{i\in T}\max\{\varepsilon_i \ , \ 0\} \ge \sum_{i\in T^*}\max\{\varepsilon_i \ , \ 0\} \ge \sum_{i\in T^*}\varepsilon_i \ge \sum_{i\in T^*}\varepsilon_i - \varepsilon^- \quad \therefore \ UB2 \ge z^*$$ ■

Upper Bound 3:

Upper Bound 3 (UB3) recognizes the use of the cycle time and the profit/time ratio of each task. We find the task that gives us the maximum profit per unit time and multiply the profit/time ratio with the cycle time by assuming that we can devote all the cycle time to the task having the maximum profit/time ratio.

$$UB3 = CT \cdot \left[\max_{i \in T} \left\{ \frac{\sum_{j \in U_i} p_j - c_i}{t_i} \right\} \right] \tag{5.6}$$

Theorem 3. UB3 is a valid upper bound on the optimal objective function value, z^*.

.

Proof.

Note that:

$UB3 = CT \cdot \varepsilon$, where $\varepsilon = \max_i \left\{ \frac{\sum_{j \in U_i} p_j - c_i}{t_i} \right\}$

Objective function value for the optimal solution can be expressed as:

$$z^* = \sum_{j \in P^*} p_j - \sum_{i \in T^*} c_i$$

We know that: $P^* \subseteq \bigcup_{i \in T^*} U_i$

$$\sum_{i \in T^*} \left(\sum_{j \in U_i} p_j - c_i \right) \geq \sum_{j \in P^*} p_j - \sum_{i \in T^*} c_i$$

Note that $\sum_{i \in T^*} \left(\sum_{j \in U_i} p_j - c_i \right) = \sum_{i \in T^*} \left(\frac{\left(\sum_{j \in U_i} p_j - c_i \right)}{t_i} \cdot t_i \right)$

Let $\varepsilon_i = \frac{\sum_{j \in U_i} p_j - c_i}{t_i}$, hence $\varepsilon = \max_i \{\varepsilon_i\}$

$$z^* = \sum_{i \in T^*} \varepsilon_i \cdot t_i \leq \varepsilon \cdot \sum_{i \in T^*} t_i \text{ as } \varepsilon \geq \varepsilon_i \ \forall i$$

$$\varepsilon \cdot \sum_{i \in T^*} t_i \leq \varepsilon \cdot CT = UB3 \text{ as } \sum_{i \in T^*} t_i \leq CT$$

Hence $UB3 \geq z^*$ ■

5.1.2 LOWER BOUND

Initial lower bound (LB) of the problem is determined by finding the most profitable part that can be produced on its own within the cycle time. If none of the parts is profitable then we take our lower bound as zero. We use the following equation for the calculation of lower bound.

$$LB = \max_j \left(p_j - \sum_{i \in S_j} c_i \ , \ 0 \right) \tag{5.7}$$

If initial LB is zero, then we assign the initial feasible solution (incumbent solution) as all $x_j = 0 \ \forall j$. If initial LB is greater then zero then we assign the incumbent solution as $x_j = 0 \ \forall j \in P \setminus \{l\}$ and $x_l = 1$, where l is the part with the maximum profit value.

5.1.3 FATHOMING

We discard the sub-problem for further consideration if its upper bound is not larger than the current best lower bound. We find initial feasible solution (initial lower bound) as previously explained and we update the current best lower bound whenever a solution with greater objective function value is found.

There is no fathoming of each B&B node due to cycle time as we initially eliminate the parts requiring recovery times greater than the remaining cycle time, by means of the size reduction mechanism.

5.1.4 THE ALGORITHM

The steps of the algorithm are as follows:

Step 1. **// Input //**

- Input DPM, PN, part revenues, task costs and times and Cycle Time (*CT*).

Step 2. **// Initial Reduction //**

- Create *PT* using DPM and PN.
- Apply Part Time Control (Reduction Rule 2) using PT and task times.
- Apply Part and Task Reduction (Reduction Rule 3) using PT, part revenues and task costs.
- Apply Task Aggregation (Reduction Rule 5) using DPM and PN.
- Apply Part Aggregation (Reduction Rule 4) using PT and part revenues.
- Create the reduced problem with new input data (PT, part revenues, task costs and task times).

Step 3. **//Formation//**

- Calculate the initial lower bound and incumbent solution accordingly.
- Form the initial problem assuming all $x_j = 0$.

Step 4. **// Reduction//**

- Calculate the remaining cycle time.
- Collect the parts with all their tasks performed.
- Apply Part Time Control (Reduction Rule 2) using PT and task times.
- Apply Part and Task Reduction (Reduction Rule 3) using PT, part revenues and task costs.

- Apply Part Aggregation (Reduction Rule 4) using PT and part revenues.
- Apply Task Insertion into Parts (Reduction Rule 6)

Step 5. **// Sub-problem //**

- Calculate the objective function, i.e., profit value of the sub-problem.
- Calculate the upper bound value of the sub-problem using UB1 and/or UB2 and/or UB3.
- If the lower bound is less than the sub-problem's objective function value then update the incumbent solution as the sub-problem and the lower bound value.
- If the upper bound value of the sub-problem is less than the lower bound then prune the branch and go to Step 7.
- If a decision for all the parts are made then go to Step 7.

Step 6. **// Branching //**

- Select the part j to be branched according to equation (5.3).
- Create two new sub-problems for $x_j = 1$ and $x_j = 0$.

Step 7. **// Node Selection //**

- Select the sub-problem to be solved using the dfs strategy and go to Step 4.
- If no sub-problems are available then STOP.

An illustrative example is presented in Appendix A.

CHAPTER 6

EXPERIMENTATION

In this chapter, we design an experiment to determine the best configuration for our branch-and-bound algorithm. We have two mechanisms for consideration: (1) upper bounds and (2) reduction rules (including the part collection). After the selection of the best combination, we conduct computational analysis of the proposed algorithm. Our aim here is to investigate the effect of some parameters on the problem difficulty. In the following sections we will discuss the experimental design and the results obtained.

6.1 EXPERIMENTAL DESIGN

We use twenty-five problems with distinct task precedence relations. Eleven of those problems are obtained from the disassembly literature. We use the power brake example (MAS30T) from Mascle et al. (2003), ball-point pen (LAM20T) and radio set (LAM30T) examples from Lambert (1999), two PC examples (GUN17T, GUN8T) from Güngör and Gupta (1997 and 2002), a coffee maker example (AKK12T) from Hula et al. (2003), weight scale example (WANG18T) of Johnson et al. (1998) and finally three artificially created examples, three from Altekin (2005) (AKO20T, AKO30T1,AKO30T2) and one from Moore et al. (1998) (MGG7T). In order to form the disassembly structure, we also reversed some of the ALBP precedence relations. A problem database is referred in Scholl (1999) and the precedence graphs of various ALBP are available in *www.wiwi.uni-jena.de/Entscheidung/alb/albdata.htm*. We selected the following problems: BOWMAN8, BUXEY, HESKIA, JACKSON, JAESCHKE, LUTZ1, MANSOOR, MERTENS, MITCHELL, ROSZIEG and SAWYER30.We also generated our own problems (YKA19T, YKA27T and YKA31T). The precedence graphs of these instances are given in 0.

In generating the instances, we first define the number of tasks and the precedence relations. For each problem, the number of parts (n) is determined by using the ratio m/n, where m is the number of tasks. We used three levels of m/n ratio, 0.50, 0.75 and 1.00 for each of the 25 problems. Therefore, we obtained 75 problem instances having distinct precedence graphs. The features of the 75 problems are presented in Table 6-1.

We set task times equal to the task costs whenever one is not available. For all problems, we generate the part revenue values using a uniform distribution with the mean and standard deviation values equivalent to the mean and standard deviation values of the task costs.

We use three different options for generating the cycle time. The cycle times are generated by multiplying the associated total task times with 0.25, 0.50 and 0.75.

So, a total of 225 problem instances are generated and solved for determining the best upper bound and the effect of the problem reduction procedures. We conduct a total of three experimental settings, each for the evaluation of one mechanism.

For each problem instance, the CPU time spent, the number of nodes and the node index where optimal solution is reached are recorded.

All the problems were solved by using Compaq Visual FORTRAN 6, on standalone HP 6280 workstations with Xeon processors, 2.8 GHz CPU and 2 GB RAM.

Table 6-1 The characteristics of the generated problems

Problem	Number of Tasks	Number of Precedence Relations	Number of Parts		
			$\frac{m}{n}=50\%$	$\frac{m}{n}=75\%$	$\frac{m}{n}=100\%$
AKK12T	12	12	6	9	12
AKO20T	15	15	8	12	15
AKO30T1	16	17	8	12	16
AKO30T2	14	16	7	11	14
GUN8T	8	9	4	6	8
GUN17T	17	24	9	13	17
LAM20T	9	8	5	7	9
LAM30T	9	8	5	7	9
MAS30T	30	32	15	23	30
MGG7T	7	7	4	6	7
WANG18T	20	19	10	15	20
YKA19T	19	20	10	15	19
YKA27T	27	32	14	21	27
YKA31T	31	42	16	24	31
BOWMAN8	8	8	4	6	8
BUXEY	29	36	15	22	29
HESKIA	28	39	14	21	28
JACKSON	11	13	6	9	11
JAESCHKE	9	11	5	7	9
LUTZ1	32	38	16	24	32
MANSOOR	11	11	6	9	11
MERTENS	7	6	4	6	7
MITCHELL	21	27	11	16	21
ROSZIEG	25	32	13	19	25
SAWYER30	30	32	15	23	30

6.2 UPPER BOUNDS

We start our experimentation for finding the most powerful upper bound, i.e., the one that result with the maximum CPU time reduction. Five different versions of the branch and bound algorithm are solved:

1. Algorithm using No Upper Bound
2. Algorithm using only Upper Bound 1, UB1
3. Algorithm using only Upper Bound 2, UB2
4. Algorithm using only Upper Bound 3, UB3
5. Algorithm using only Best Upper Bound,

UB_BEST=Min{UB1,UB2,UB3}

The CPU times and total number of nodes in the B&B tree and the indices of the optimal nodes for each version, are presented in Table 6-2 and Table 6-3 respectively for cycle time 0.5 of total task time and *m/n* ratio 0.75. The complete results including the remaining factor levels are presented in Appendix C.

As can be observed from Table 6-2 and Table 6-3, UB2 is the most efficient upper bound, leading to the most significant improvement in solution times. When the number of nodes and the node index at which optimal solution is reached are the concern, the most effective upper bound is UB_BEST followed with UB2. UB2 and UB_BEST are followed by UB1 for the CPU times and the number of B&B nodes. UB3 is the worst upper bound, producing very poor results when compared to other upper bounds.

In order to reach a conclusion from all the results for 225 cases, we perform additional analysis. In terms of CPU times for each problem, we ranked all the upper bound solutions from minimum to maximum (rank 1 represents the minimum CPU incurred, i.e. maximum reduction obtained for the specific problem instance) for each 225 problem instance. Afterwards we calculate the percentage of those rankings for each upper bound. Table 6-4 presents the associated results. From Table 6-4, we can conclude that the results we obtain from Table 6-3 are valid when all the problem instances are analyzed.

Table 6-2 B&B node results for the initial upper bound experimentation (*CT*=50%, *m/n*=75%)

			NO UB		UB 1		UB 2		UB 3		UB_BEST	
Test Problem	***n***	***m***	**# of Nodes**	**Optimal Node Index**	**# of Nodes**	**Optimal Node Index**	**# of Nodes**	**Optimal Node Index**	**# of Nodes**	**Optimal Node Index**	**# of Nodes**	**Optimal Node Index**
BOWMAN8-2-0	8	6	15	3	7	3	7	3	15	3	7	3
BUXEY-2-0	29	22	131071	31	2437	31	203	31	70921	31	177	31
MERTENS-2-0	7	6	31	3	17	3	17	3	17	3	17	3
HESKIA-2-0	28	21	2097151	23	433	23	185	23	1636749	23	177	23
JACKSON-2-0	11	9	511	11	439	11	23	11	367	11	19	11
JAESCHKE-2-0	9	7	15	3	9	3	9	3	11	3	9	3
LUTZ1-2-0	32	24	65535	11	8615	11	499	11	36767	11	349	11
MANSOOR-2-0	11	9	511	15	121	15	19	15	367	15	19	15
MITCHELL-2-0	21	16	4095	21	2417	21	67	21	2777	21	57	21
ROSZIEG-2-0	25	19	16383	19	173	19	113	19	14579	19	101	19
SAWYER30-2-0	30	23	524287	25	1223	25	195	25	289971	25	145	25
AKKSKT12-2-0	12	9	511	11	239	11	23	11	383	11	17	11
AKO20T-2-0	15	12	2047	1539	435	377	83	65	1227	919	75	57
AKO30T1-2-0	16	12	2047	11	1339	11	69	11	1441	11	57	11
AKO30T2-2-0	14	11	1023	13	243	13	29	13	769	13	23	13
GUN17T-2-0	17	13	4095	15	349	15	43	15	2611	15	25	15
GUN8T-2-0	8	6	7	5	7	5	5	5	7	5	5	5
LAM20T-2-0	9	7	255	9	27	9	21	9	139	9	15	9
LAM30T-2-0	9	7	31	3	11	3	9	3	21	3	9	3
MAS30T-2-0	30	23	2097151	12821	553	127	825	179	1230407	11831	455	89
MGG7T-2-0	7	6	63	9	51	9	13	9	41	9	11	9
WANG18T-2-0	20	15	65535	13	251	13	291	13	65497	13	157	13
YKA19T-2-0	19	15	65535	13	567	13	159	13	15231	13	129	13
YKA27T-2-0	27	21	4194303	23	2091	23	457	23	639177	23	357	23
YKA31T-2-0	31	24	16777216	31	6973	31	877	31	11690531	31	641	31

Table 6-3 CPU times results for the initial upper bound experimentation (*CT*=50%, *m/n*=75%)

			NO UB	UB 1	UB 2	UB 3	UB_BEST
Test Problem	***n***	***m***	**CPU Time**	**CPU Time**	**CPU Time**	**CPU Time**	**CPU Time**
BOWMAN8-2-0	8	6	0.0000	0.0000	0.0000	0.0160	0.0000
BUXEY-2-0	29	22	26.3280	0.6400	0.0630	16.8750	0.1100
MERTENS-2-0	7	6	0.0000	0.0000	0.0000	0.0000	0.0000
HESKIA-2-0	28	21	392.7720	0.1400	0.0470	372.2820	0.1250
JACKSON-2-0	11	9	0.0160	0.0310	0.0000	0.0160	0.0150
JAESCHKE-2-0	9	7	0.0000	0.0000	0.0000	0.0000	0.0000
LUTZ1-2-0	32	24	15.2300	2.5040	0.1720	10.0490	0.2350
MANSOOR-2-0	11	9	0.0160	0.0150	0.0000	0.0160	0.0000
MITCHELL-2-0	21	16	0.4680	0.3270	0.0160	0.3740	0.0150
ROSZIEG-2-0	25	19	2.5400	0.0470	0.0150	2.6970	0.0470
SAWYER30-2-0	30	23	115.0090	0.3900	0.0620	76.4810	0.0930
AKKSKT12-2-0	12	9	0.0160	0.0150	0.0000	0.0310	0.0000
AKO20T-2-0	15	12	0.1410	0.0470	0.0000	0.1090	0.0160
AKO30T1-2-0	16	12	0.1410	0.1250	0.0000	0.1250	0.0150
AKO30T2-2-0	14	11	0.0620	0.0160	0.0000	0.0620	0.0000
GUN17T-2-0	17	13	0.3280	0.0470	0.0000	0.2650	0.0150
GUN8T-2-0	8	6	0.0000	0.0000	0.0000	0.0000	0.0000
LAM20T-2-0	9	7	0.0000	0.0160	0.0000	0.0000	0.0000
LAM30T-2-0	9	7	0.0000	0.0000	0.0000	0.0000	0.0000
MAS30T-2-0	30	23	467.7650	0.1880	0.2810	327.9740	0.3430
MGG7T-2-0	7	6	0.0000	0.0000	0.0160	0.0000	0.0000
WANG18T-2-0	20	15	7.2030	0.0320	0.0620	8.7500	0.0620
YKA19T-2-0	19	15	7.0780	0.0940	0.0160	2.1090	0.0470
YKA27T-2-0	27	21	823.7550	0.6720	0.1410	153.1260	0.2340
YKA31T-2-0	31	24	4271.6520	2.7970	0.3750	3633.0710	0.5780

Table 6-4 Results of the initial upper bound evaluation runs

%	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
No UB	0	0	1	40	59
UB 1	12*	18	66	4	0
UB 2	78	14	7	1	0
UB 3	1	0	4	55	40
UB_BEST	9	68	22	0	1

* For example: Upper Bound 1 is ranked first among five combinations in CPU time in 12% of the test cases.

We also performed additional runs by using UB2, UB_BEST and Min{UB1,UB2}. The results associated to those runs are presented in Table 6-5. Table 6-5 contains the results for cycle time 50% and *m/n* ratio 0.75. The complete results including the remaining factor levels are presented in Appendix C. Results shows that UB2 outperforms UB_BEST and Min{UB1,UB2}.

Note that the number of nodes in the B&B tree with UB_BEST and Min{UB1,UB2} is smaller when compared with UB2. However, such a reduction cannot be justified due to increase in the CPU times. The CPU times by UB2 are the smallest even whenever the number of nodes is highest. The results have also revealed that, when the problem size gets larger, the CPU time difference between UB2 and the other bound combinations are much more noticeable in the favor of UB2.

By this analysis, we confirm our findings over all problem set. When CPU time are considered, for 60.2% of the problem cases UB2 performed better, followed by UB_BEST with 23.3% and Min{UB1, UB2} with 16.5%. We also observed that when the problem size is small, the differences between the performances are not very significant as expected; the difference between the upper bound becomes more apparent with an increase in the number of parts due to our branching scheme that settles one part decision at each level.

Table 6-5 B&B results for the detailed upper bound experimentation (*CT*=50%, *m/n*=75%)

			UB 2				UB_BEST				BEST OF UB 1 UB 2			
Test Problem	***n***	***m***	**# of Nodes**	**Opt. Node Index**	**CPU Time**	**Max CPU Time**	**# of Nodes**	**Opt. Node Index**	**CPU Time**	**Max CPU Time**	**# of Nodes**	**Opt. Node Index**	**CPU Time**	**Max CPU Time**
BOWMAN8-2-0	8	6	7	3	0.0000	0.0000	7	3	0.0000	0.0000	7	3	0.0047	0.0160
BUXEY-2-0	29	22	203	31	0.0640	0.0780	177	31	0.1137	0.1250	2437	31	0.3227	0.3290
MERTENS-2-0	7	6	17	3	0.0000	0.0000	17	3	0.0000	0.0000	17	3	0.0000	0.0000
HESKIA-2-0	28	21	185	23	0.0563	0.0630	177	23	0.1199	0.1250	739	23	0.1261	0.1400
JACKSON-2-0	11	9	23	11	0.0046	0.0160	19	11	0.0047	0.0160	99	11	0.0079	0.0160
JAESCHKE-2-0	9	7	9	3	0.0016	0.0160	9	3	0.0000	0.0000	9	3	0.0000	0.0000
LUTZ1-2-0	32	24	499	11	0.1700	0.1720	349	11	0.2292	0.2340	8615	11	1.2512	1.2780
MANSOOR-2-0	11	9	19	15	0.0032	0.0160	19	15	0.0031	0.0160	21	15	0.0016	0.0160
MITCHELL-2-0	21	16	67	21	0.0124	0.0160	57	21	0.0187	0.0310	433	21	0.0344	0.0470
ROSZIEG-2-0	25	19	113	19	0.0264	0.0320	101	19	0.0469	0.0470	243	19	0.0281	0.0320
SAWYER30-2-0	30	23	195	25	0.0686	0.0780	145	25	0.0999	0.1090	1169	25	0.1856	0.1880
AKKSKT12-2-0	12	9	23	11	0.0015	0.0150	17	11	0.0047	0.0160	41	11	0.0032	0.0160
AKO20T-2-0	15	12	83	65	0.0109	0.0160	75	57	0.0158	0.0160	107	83	0.0079	0.0160
AKO30T1-2-0	16	12	69	11	0.0032	0.0160	57	11	0.0122	0.0160	301	11	0.0191	0.0310
AKO30T2-2-0	14	11	29	13	0.0000	0.0000	23	13	0.0032	0.0160	195	13	0.0110	0.0160
GUN17T-2-0	17	13	43	15	0.0032	0.0160	25	15	0.0125	0.0160	871	15	0.0472	0.0520
GUN8T-2-0	8	6	5	5	0.0015	0.0150	5	5	0.0000	0.0000	7	5	0.0015	0.0150
LAM20T-2-0	9	7	21	9	0.0015	0.0150	15	9	0.0031	0.0160	21	9	0.0000	0.0000
LAM30T-2-0	9	7	9	3	0.0016	0.0160	9	3	0.0030	0.0150	11	3	0.0015	0.0150
MAS30T-2-0	30	23	825	179	0.2823	0.2960	455	89	0.3382	0.3430	543	119	0.0977	0.1090
MGG7T-2-0	7	6	13	9	0.0000	0.0000	11	9	0.0015	0.0150	13	9	0.0000	0.0000
WANG18T-2-0	20	15	291	13	0.0515	0.0630	157	13	0.0560	0.0630	299	13	0.0237	0.0320
YKA19T-2-0	19	15	159	13	0.0279	0.0310	129	13	0.0373	0.0470	1021	13	0.0804	0.0940
YKA27T-2-0	27	21	457	23	0.1464	0.1560	357	23	0.2372	0.2500	1529	23	0.2509	0.2800
YKA31T-2-0	31	24	877	31	0.3618	0.3750	641	31	0.5752	0.5770	6795	31	1.3817	1.5180

6.3 REDUCTION MECHANISMS

In order to evaluate the performance of the reduction mechanisms (including the part collection), we use the same experimental setting. We try two versions of the B&B algorithm: one using reduction mechanisms, the other not using, and present the results in Table 6-7 (Table 6-7 contains the results for cycle time 50% and *m/n* ratio 100%, complete results including the remaining factor levels are presented in Appendix C). For all problem instances, utilization of reduction mechanisms resulted in better solutions when the total number of nodes in the B&B tree is considered. When CPU time is the concern it can be observed that the results are close yet reduction mechanisms is solving the problem with lower CPU time values. Utilization of the reduction mechanisms yields better CPU times for 94% of the problems. From the results we can conclude that the reduction mechanisms play an important role in reducing the number of nodes and the CPU times.

The overall average reduction in the size of the B&B tree and CPU times are 53.7% and 58.8% respectively. Table 6-6 reports the average reduction percentages that can be obtained by using the reduction mechanisms in the total number of nodes and the CPU times for different cycle time levels. The percentage reductions in the number of nodes and the CPU times increase with an increase in the cycle times.

Table 6-6 Average Percent reductions in number of nodes and CPU times by utilization of the reduction mechanisms

Cycle Time as a % of Total Task Times	Number Of B&B Nodes	CPU Time
25	38.5%	52.7%
50	58.2%	60.4%
75	64.5%	61.2%

Table 6-7 B&B results for the reduction mechanisms experimentation (*CT*=50%, *m/n*=100%)

			REDUCTION				NO REDUCTION			
Test Problem	***n***	***m***	**# of Nodes**	**Optimal Node Index**	**CPU Time**	**Max CPU Time**	**# of Nodes**	**Optimal Node Index**	**CPU Time**	**Max CPU Time**
BOWMAN8-3-0	8	8	9	5	0.0015	0.0150	15	5	0.0015	0.0150
BUXEY-3-0	29	29	179	27	0.2109	0.2190	397	31	0.1714	0.1720
MERTENS-3-0	7	7	3	0	0.0000	0.0000	11	1	0.0046	0.0160
HESKIA-3-0	28	28	29	29	0.0421	0.0470	2081	53	0.9229	0.9350
JACKSON-3-0	11	11	15	15	0.0031	0.0160	43	17	0.0048	0.0160
JAESCHKE-3-0	9	9	9	5	0.0000	0.0000	19	5	0.0015	0.0150
LUTZ1-3-0	32	32	159	31	0.1560	0.1720	3929	31	1.9020	1.9020
MANSOOR-3-0	11	11	19	19	0.0046	0.0160	23	19	0.0016	0.0160
MITCHELL-3-0	21	21	129	23	0.0781	0.0790	323	23	0.0749	0.0780
ROSZIEG-3-0	25	25	191	25	0.1610	0.1870	659	27	0.2137	0.2190
SAWYER30-3-0	30	30	229	57	0.4159	0.4220	1087	121	0.5241	0.5310
AKKSKT12-3-0	12	12	19	11	0.0016	0.0160	51	11	0.0047	0.0160
AKO20T-3-0	15	15	9	0	0.0000	0.0000	145	1	0.0186	0.0310
AKO30T1-3-0	16	16	27	21	0.0108	0.0160	95	31	0.0159	0.0160
AKO30T2-3-0	14	14	11	11	0.0030	0.0150	59	31	0.0032	0.0160
GUN17T-3-0	17	17	35	15	0.0062	0.0160	75	19	0.0109	0.0160
GUN8T-3-0	8	8	5	5	0.0000	0.0000	11	5	0.0000	0.0000
LAM20T-3-0	9	9	5	3	0.0000	0.0000	45	3	0.0031	0.0160
LAM30T-3-0	9	9	9	5	0.0016	0.0160	21	7	0.0016	0.0160
MAS30T-3-0	30	30	419	61	0.4828	0.4850	6233	291	2.9888	2.9940
MGG7T-3-0	7	7	7	7	0.0000	0.0000	13	11	0.0000	0.0000
WANG18T-3-0	20	20	179	21	0.1798	0.1880	2329	29	0.5442	0.5460
YKA19T-3-0	19	19	53	19	0.0421	0.0470	595	25	0.1311	0.1410
YKA27T-3-0	27	27	21	17	0.0141	0.0160	1885	23	0.7782	0.7800
YKA31T-3-0	31	31	173	35	0.4453	0.4690	3927	41	2.1640	2.1670

6.4 FURTHER ANALYSIS

Our initial experiments have revealed that using UB2 together with the reduction mechanisms results in the best performance. Hence, we create a bigger test bed and analyze the computational efficiency of our algorithm that uses UB2 and reduction mechanisms.

We create our new test bed by random generation of 9 additional sets of part data for each of 225 problem instances. We keep the task times and costs fixed and randomly change the part revenues for each problem described in Section 6.1. This yields a total of 2250 problem instances.

Table 6-8, Table 6-9 and Table 6-10 report the total number of nodes in the B&B tree and CPU times for the runs (Tables contains the results for *m/n* ratio 75%, complete results including the remaining factor levels are presented in Appendix C). As can be observed from the tables, the CPU times increase along with increasing number of parts, (*m*), and number of tasks, (*n*). From Figure 6.1 and Figure 6.2 (Figures represent the all 2250 problems) we can observe that the CPU times increase with an increase polynomial in the number of parts and tasks. Note that, each level of our B&B tree sets one decision: either a part is produced or not. There are a total of 2^m complete solutions that will be implicitly enumerated by our algorithm. According to our decision structure, the size of the search is dependent on the number of parts, (*m*), rather than the number of tasks, (*n*). The decisions on the task selections are dependent on part selections. Hence, one should expect an exponential increases of the CPU times and number of nodes with an increase in (*m*). But, in our case, we obtain a polynomial increase which can be attributed to our powerful bounding mechanisms.

Table 6-8 B&B results for computational analysis (*CT*=25%, *m/n*=75%)

Test Problem	*n*	*m*	Avg. # of Nodes	Max # of Nodes	Avg. Optimal Node Index	Max Optimal Node	Avg. Opt. Node/ Node	Avg. CPU Time	Max CPU Time
BOWMAN8-2	8	6	5.8	9	5.2	9	0.9206	0.0020	0.0160
BUXEY-2	29	22	15	27	9	17	0.6542	0.0037	0.0160
MERTENS-2	7	6	6.2	13	3.2	7	0.5691	0.0019	0.0160
HESKIA-2	28	21	24	29	23.8	27	0.9931	0.0098	0.0320
JACKSON-2	11	9	5.6	9	4.8	9	0.8514	0.0014	0.0160
JAESCHKE-2	9	7	5.6	7	5	7	0.8800	0.0015	0.0160
LUTZ1-2	32	24	23	45	14.6	25	0.6960	0.0078	0.0310
MANSOOR-2	11	9	3.4	5	2.8	5	0.8000	0.0017	0.0160
MITCHELL-2	21	16	11.8	23	7.6	13	0.7252	0.0027	0.0160
ROSZIEG-2	25	19	15.4	27	12	21	0.8397	0.0038	0.0160
SAWYER30-2	30	23	19.4	45	11	27	0.5504	0.0060	0.0310
AKKSKT12-2	12	9	8.8	11	8.8	11	1.0000	0.0021	0.0160
AKO20T-2	15	12	5	11	5	11	1.0000	0.0027	0.0160
AKO30T1-2	16	12	12	21	11	19	0.9023	0.0030	0.0160
AKO30T2-2	14	11	4.6	11	3.6	11	0.8076	0.0030	0.0160
GUN17T-2	17	13	14.6	19	13.8	17	0.9474	0.0031	0.0160
GUN8T-2	8	6	3	5	3	5	1.0000	0.0005	0.0160
LAM20T-2	9	7	3.8	5	3.8	5	1.0000	0.0009	0.0160
LAM30T-2	9	7	5.2	7	5.2	7	1.0000	0.0020	0.0160
MAS30T-2	30	23	26.2	57	21	27	0.8824	0.0111	0.0470
MGG7T-2	7	6	3.6	7	2.8	5	0.8476	0.0009	0.0160
WANG18T-2	20	15	14.6	19	13.4	19	0.9279	0.0041	0.0160
YKA19T-2	19	15	16.6	23	12.2	17	0.7303	0.0058	0.0320
YKA27T-2	27	21	29.2	51	18.4	23	0.6721	0.0227	0.0470
YKA31T-2	31	24	41.8	77	21	47	0.5682	0.0321	0.0630

Table 6-9 B&B results for computational analysis (*CT*=50%, *m/n*=75%)

Test Problem	*n*	*m*	Avg. # of Nodes	Max # of Nodes	Avg. Optimal Node Index	Max Optimal Node	Avg. Opt. Node/ Node	Avg. CPU Time	Max CPU Time
BOWMAN8-2	8	6	6.8	11	6.2	11	0.9206	0.0014	0.0160
BUXEY-2	29	22	62.4	81	25.4	29	0.4203	0.0352	0.0630
MERTENS-2	7	6	7.2	13	3.4	9	0.5296	0.0008	0.0160
HESKIA-2	28	21	23.6	27	23.6	27	1.0000	0.0162	0.0320
JACKSON-2	11	9	15.2	21	11.2	17	0.7444	0.0037	0.0160
JAESCHKE-2	9	7	7.8	9	7.2	9	0.9156	0.0011	0.0160
LUTZ1-2	32	24	120.6	303	23.2	29	0.4102	0.0641	0.2030
MANSOOR-2	11	9	12.4	17	12.2	15	0.9882	0.0028	0.0160
MITCHELL-2	21	16	31	63	18.8	25	0.7236	0.0110	0.0320
ROSZIEG-2	25	19	67.6	173	15	27	0.3521	0.0346	0.1100
SAWYER30-2	30	23	95.6	183	27.6	77	0.3411	0.0974	0.2500
AKKSKT12-2	12	9	11.8	27	9	11	0.8559	0.0014	0.0160

Table 6-9 B&B results for computational analysis (CT=50%, m/n=75%) (Continued)

Test Problem	*n*	*m*	Avg. # of Nodes	Max # of Nodes	Avg. Optimal Node	Max Optimal Node	Avg. Opt. Node/ Node	Avg. CPU Time	Max CPU Time
AKO20T-2	15	12	20.8	45	13	31	0.6907	0.0058	0.0160
AKO30T1-2	16	12	21.6	27	18.2	21	0.8534	0.0046	0.0160
AKO30T2-2	14	11	20.8	29	16.8	25	0.8195	0.0036	0.0160
GUN17T-2	17	13	26.8	55	15.6	25	0.6710	0.0076	0.0160
GUN8T-2	8	6	4.4	7	4	7	0.8933	0.0010	0.0160
LAM20T-2	9	7	12	15	10	15	0.8225	0.0027	0.0160
LAM30T-2	9	7	8.6	13	7.2	11	0.8537	0.0015	0.0160
MAS30T-2	30	23	165.4	295	36.6	61	0.2614	0.1302	0.3750
MGG7T-2	7	6	6	11	4.6	9	0.8416	0.0014	0.0160
WANG18T-2	20	15	38.8	89	14	19	0.4954	0.0152	0.0470
YKA19T-2	19	15	23.6	57	13.2	19	0.6589	0.0108	0.0320
YKA27T-2	27	21	27	51	18.4	23	0.7332	0.0224	0.0470
YKA31T-2	31	24	230	657	25	49	0.2745	0.3862	1.0630

Table 6-10 B&B results for computational analysis (*CT*=75%, *m/n*=75%)

Test Problem	*n*	*m*	Avg. # of Nodes	Max # of Nodes	Avg. Optimal Node Index	Max Optimal Node	Avg. Opt. Node/ Node	Avg. CPU Time	Max CPU Time
BOWMAN8-2	8	6	7.8	11	6.2	11	0.7757	0.0014	0.0160
BUXEY-2	29	22	381.8	647	38.6	69	0.1215	0.6587	1.1410
MERTENS-2	7	6	7.8	15	3.4	9	0.5138	0.0013	0.0160
HESKIA-2	28	21	23.6	27	23.6	27	1.0000	0.0155	0.0320
JACKSON-2	11	9	15.2	21	11.2	17	0.7444	0.0025	0.0160
JAESCHKE-2	9	7	10.8	13	7.8	13	0.7568	0.0016	0.0160
LUTZ1-2	32	24	349	961	26.6	39	0.2366	0.3463	0.9370
MANSOOR-2	11	9	12.4	17	12.2	15	0.9882	0.0003	0.0160
MITCHELL-2	21	16	50.6	133	19	25	0.5520	0.0248	0.0630
ROSZIEG-2	25	19	121	297	15	27	0.2674	0.0851	0.2340
SAWYER30-2	30	23	246.2	441	37	143	0.2320	0.4326	0.8910
AKKSKT12-2	12	9	11.8	27	9	11	0.8559	0.0023	0.0160
AKO20T-2	15	12	42.2	89	20.6	65	0.4960	0.0141	0.0320
AKO30T1-2	16	12	21.6	27	18.2	21	0.8534	0.0050	0.0160
AKO30T2-2	14	11	20.8	29	16.8	25	0.8195	0.0050	0.0160
GUN17T-2	17	13	46.2	111	19.8	39	0.5748	0.0163	0.0470
GUN8T-2	8	6	6.6	11	6.4	11	0.9714	0.0006	0.0160
LAM20T-2	9	7	12	15	10	15	0.8225	0.0014	0.0160
LAM30T-2	9	7	9.2	13	8	11	0.8775	0.0014	0.0160
MAS30T-2	30	23	578.4	1819	61.2	183	0.2212	0.9863	4.7030
MGG7T-2	7	6	6	11	4.6	9	0.8416	0.0011	0.0160
WANG18T-2	20	15	38.8	89	14	19	0.4954	0.0157	0.0470
YKA19T-2	19	15	23.6	57	13.2	19	0.6589	0.0101	0.0320
YKA27T-2	27	21	27	51	18.4	23	0.7332	0.0217	0.0470
YKA31T-2	31	24	230	657	25	49	0.2745	0.3852	1.0470

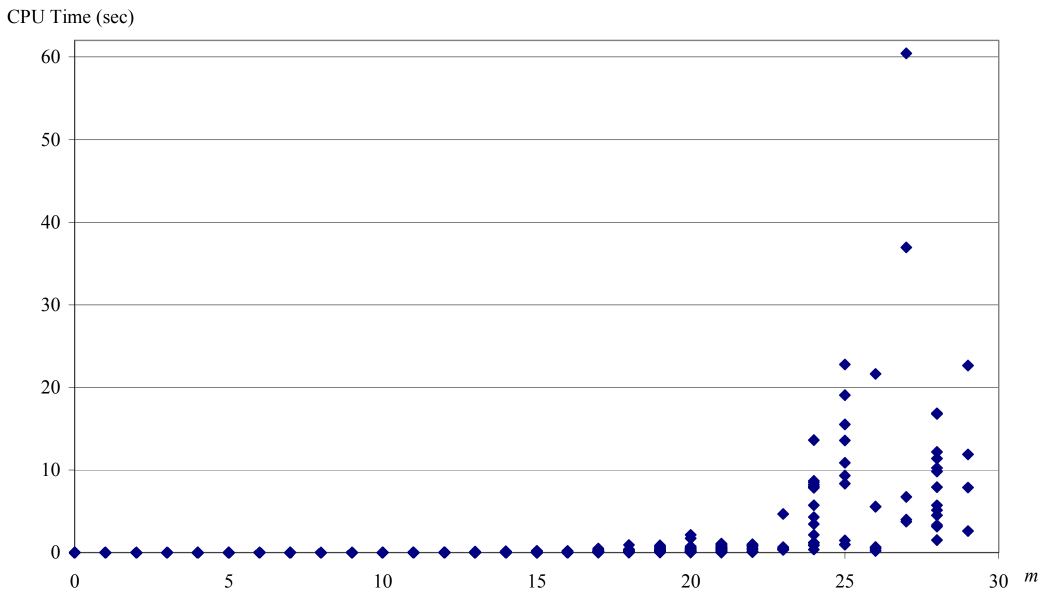


Figure 6.1 CPU time versus number of parts

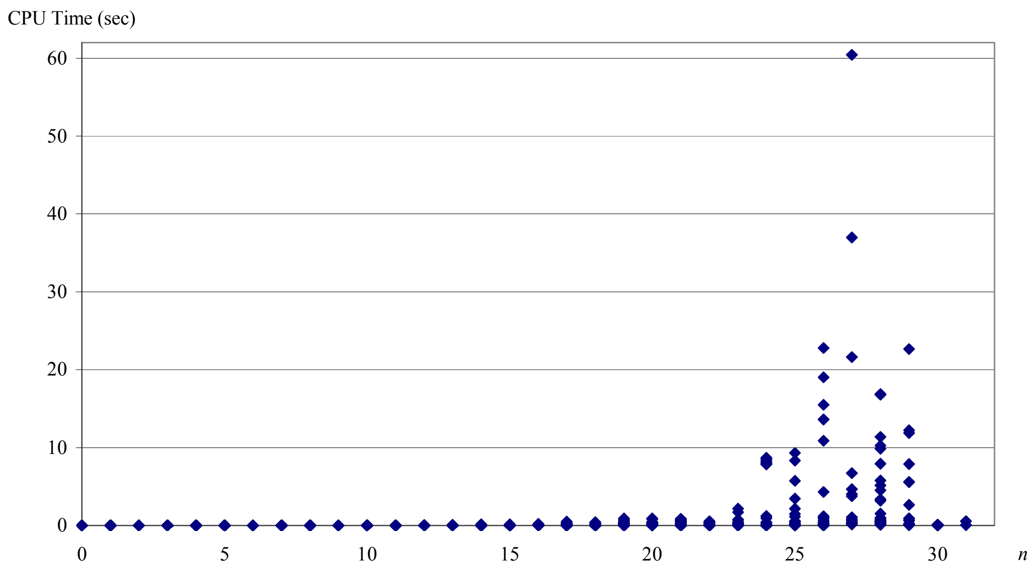


Figure 6.2 CPU time versus number of tasks

Same pattern of increase in CPU times can also be observed for the number of nodes created (Figure C.1 and Figure C.2). Thus, CPU time linearly increases with the increasing number of nodes as it can be observed from

Table 6-8, Table 6-9, and Table 6-10 (Figure C.3 also presents the results for the whole 2250 problem instances). The amount of time spent on each sub-problem depends on both number of tasks and number of parts, as *m* and *n* increases the time necessary to deal with a single sub-problem also increases as computations become more involved. So it can be concluded that as problem size increases the number of nodes and the process time of each node increases.

The efficiency of the bounding mechanism can be observed from

Table 6-8, Table 6-9 and Table 6-10. In the tables, we present the ratio of the optimal node index to the number of nodes in the B&B tree. Figure 6.4 shows the change in the ratio with respect to the total number of B&B nodes. The ratio decreases as the number of nodes increases. This is due to the efficiency of the bounding and reduction mechanisms in guiding the search. The average value obtained over all problem instances is 0.68. As our problem size gets larger the ratio gets smaller. The over all maximum ratio we observed is 1 and the minimum ratio is 0.0046 belonging to our largest sized problem, LUTZ1.

We also report the node number at which optimal solution is reached. The relation between the optimal node number and *m* can be seen in Figure 6.3 for the whole test set. The trend of increase in the number of the optimal node is steeper than the change in the number of nodes. (The maximum number of optimality node observed is 963 for a problem with 29 parts. A significant pattern cannot be observed for the optimal node. We can only state that the trend is an ascending one with the maximum value observed remaining below 1000.)

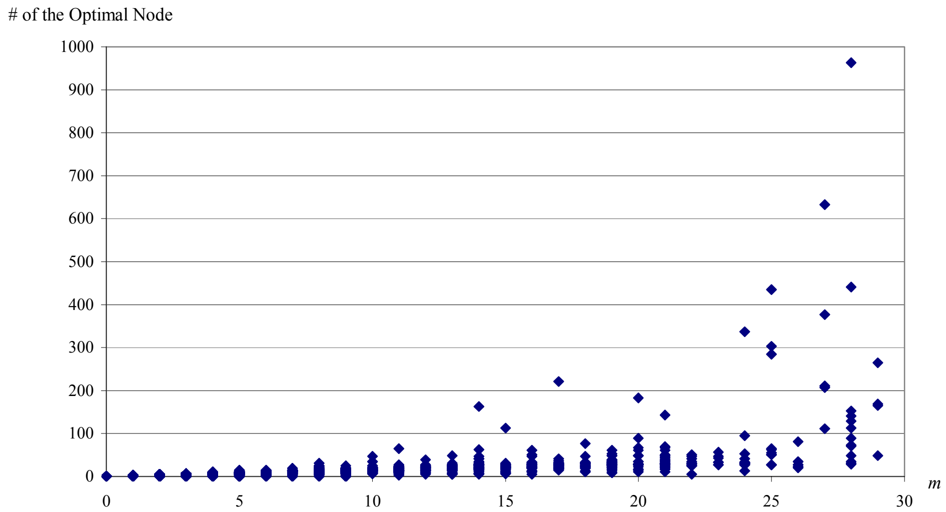


Figure 6.3 Number of the optimal node versus number of parts

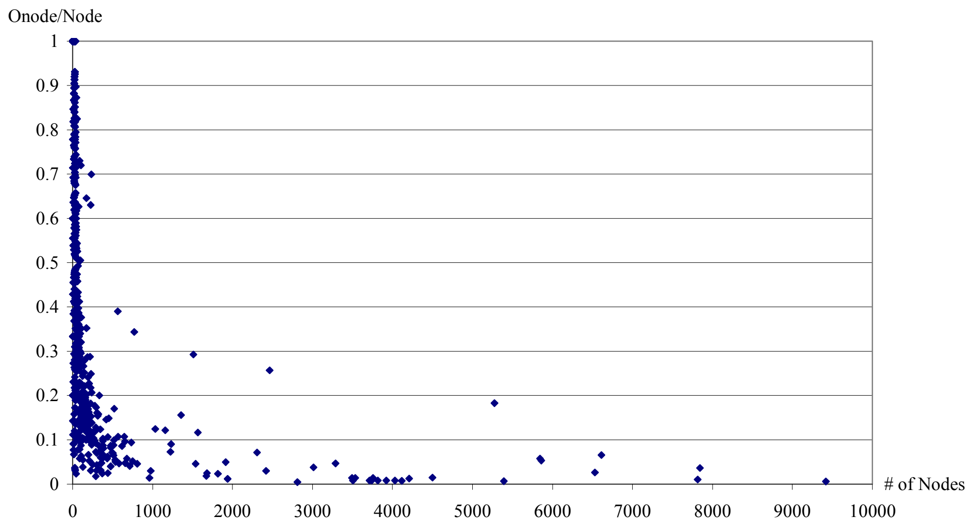


Figure 6.4 Optimal node index/ number of nodes ratio versus number of nodes

To discuss the effects of the parameters cycle time and *m/n* ratio on the total number of nodes, CPU time and optimal node index ratio, we present Table 6-11, 6-12 and 6-13 respectively.

Table 6-11 Number of B&B nodes with changing cycle time and *m/n* ratio

		Cycle Time as a % of Total Task Times					
		25%		50%		75%	
		Average	Max	Average	Max	Average	Max
m/n	0.50	8.24	33	16.368	153	22.944	225
	0.75	12.968	77	42.712	657	92.016	1819
	1.00	21.008	385	102.016	1937	741.352	19615

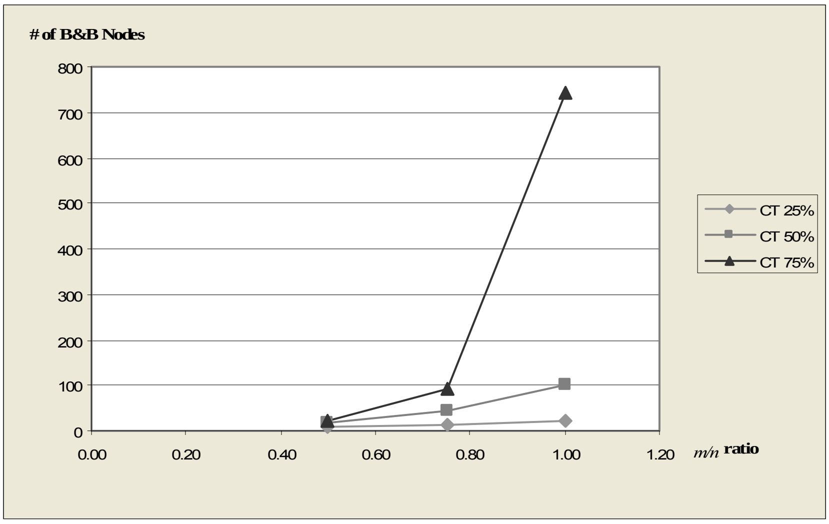


Figure 6.5 Number of B&B nodes versus *m/n* ratio

Table 6-12 CPU time with changing cycle time and *m/n* ratio

		Cycle Time as a % of Total Task Times					
		25%		50%		75%	
		Average	Max	Average	Max	Average	Max
m/n	0.50	0.00311	0.032	0.00639	0.156	0.01285	0.235
	0.75	0.00545	0.063	0.0345	1.063	0.12144	4.703
	1.00	0.01322	0.578	0.13914	5.625	1.93981	60.454

Table 6-13 Optimal node index ratio with changing cycle time and *m/n* ratio

		Cycle Time as a % of Total Task Times		
		25%	50%	75%
		Average	Average	Average
m/n	0.50	0.92035	0.79737	0.75125
	0.75	0.83063	0.68283	0.62737
	1.00	0.74298	0.5872	0.52102

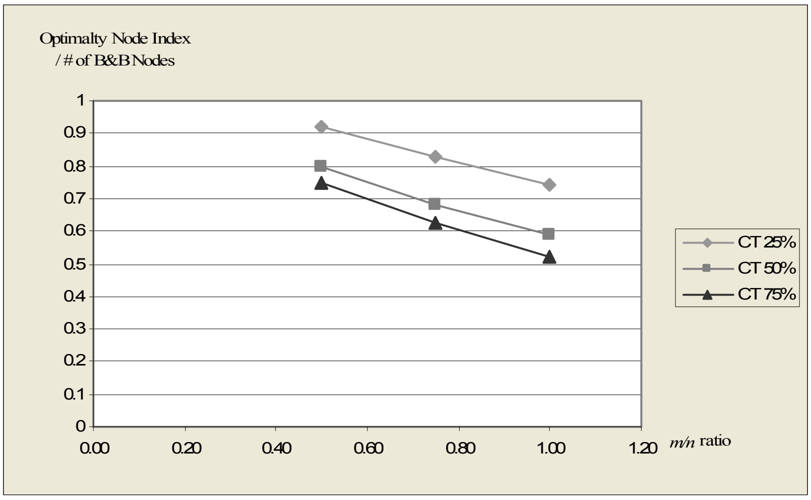


Figure 6.6 Optimality node index ratio versus *m/n* ratio

From Table 6-11 we observe that, as expected, number of B&B nodes increases with an increase in both cycle time and *m/n* ratio. When we increase the cycle time the number of feasible solutions increase so the search space increases. When we increase the *m/n* ratio while keeping the cycle time constant, we increase the number of parts that can be recovered with in the given cycle time. Again we increase the number of feasible solutions and thus the search space. By changing both cycle time and m/n ratio we increase the number of parts that can be recovered with in the given cycle time. CPU time is closely dependent on the number of nodes so the cycle time and *m/n* ratio affect the number of nodes similarly.

Upper bounds are also affected by the changes in problem parameters. As can be observed from equation (5.5) our upper bound representation is directly affected from the increase in the number of parts and task. When we increase the cycle time we increase both the number of fittable parts, but we also increase the number of tasks. Yet when *m/n* ratio is increased we only increase the number of fittable parts. Thus we can say that the effect of increasing the cycle time on upper bound performance is more than the effect of increasing the *m/n* ratio.

The effect can be observed for Table 6-11 and Table 6-12 for cycle time levels 25% and 50 %. Table 6-17 presents the effect cycle time levels on the performance. From Table 6-17, we can observe that the optimal objective function values are close for cycle time levels 50% and 75%. These cases also have the same number of B&B nodes and optimality node indices. Thus the effects of cycle time and m/n ratio on upper bound performances can not be observed from Table 6-11 and Table 6-12 for cycle time levels 50% and 75%.

For the all set we have 750 problem instances each solved for all the three cycle time levels (a total of 2250 instances). When cycle time level 1 (25%) is increased to 2 (50%), we observe an average increase of 48% in the number of B&B nodes, over the 750 cases. However, we only observe an increase in the objective function values for 401 cases. We observe that the optimal solution value obtained for cycle time level 2 (50%) is equal to the optimal solution value obtained for cycle time level 3 (75%) for 681 cases out of 750. For 515 among the 681 cases, we observe that the number of nodes is not affected by the cycle time levels. Thus, when cycle time effect is our concern, for every problem instance solved, we could not observe the behavior in Table 6-11 for cycle time levels 50% and 75%. But the effects of *m/n* ratios have the same tendency with Table 6-11 for most of the problem instances solved (71 out of 75 instances).

Table 6-13 and Figure 6.6 present the effects of cycle time and *m/n* ratio on the optimal node index ratio. It can be observed that as the number of feasible solutions increase the optimality node index ratio decreases. This result is expected because due to the branching and reduction mechanisms the optimal solution is found in early nodes.

To analyze the change in the optimal solution values and solution performance measures, we performed some analyze by solving Problem I and Problem II together. In this analysis, we solve Problem I to determine the maximum profit that can be obtained in the absence of cycle time constraint. Then we solve Problem II by using 75% of the previously determined disassembly time as the cycle time. The results are presented in Table 6-18.

The average reduction observed in the optimal solution value, when the cycle time is limited to 75% of the maximum profit case, is 19%. Hence a 25% reduction in the optimal solution time resulted in 19% reduction in the objective function value. We have observed 55% reduction for the number of nodes and 48.6% reduction in the CPU times.

The effects of the cycle time and *m/n* ratio on the total number of nodes, CPU time and optimal node index ratio are presented in Table 6-14, 6-15 and 6-16 respectively. The affects of *m/n* ratio and cycle time on upper bound performances can be easily observed from the tables.

Table 6-14 Number of B&B nodes for make span analysis

		Problem I		**Problem II**	
		Average	**Max**	**Average**	**Max**
m/n	**0.50**	35.8	194	11.6	36.8
	0.75	180.4	1551.4	30.9	123.4
	1.00	1687.8	20184.6	63.6	338.8

Table 6-15 CPU time for make span analysis

		Problem I		Problem II	
		Average	Max	Average	Max
m/n	**0.50**	0.024	0.095	0.010	0.039
	0.75	0.1570	1.634	0.030	0.18
	1.00	2.738	38.315	0.090	0.627

Table 6-16 Optimal node index ratio for make span analysis

		Problem I	Problem II
		Average	Average
m/n	**0.50**	0.585	0.859
	0.75	0.470	0.708
	1.00	0.382	0.619

Table 6-17 B&B results for cycle time effect analysis (*m/n*=100%)

			CYCLE TIME LEVEL 1 %25			CYCLE TIME LEVEL 2 %50			CYCLE TIME LEVEL 3 %75		
Test Problem	***n***	***m***	**# of Nodes**	**Optimal Node Index**	**Optimal Solution Value**	**# of Nodes**	**Optimal Node Index**	**Optimal Solution Value**	**# of Nodes**	**Optimal Node Index**	**Optimal Solution Value**
BOWMAN8-3	8	8	7	5	9	9	5	9	13	13	10
BUXEY-3	29	29	19	9	9	179	27	19	1673	31	22
MERTENS-3	7	7	3	1	0	3	1	0	1	1	0
HESKIA-3	28	28	29	29	275	29	29	275	29	29	275
JACKSON-3	11	11	5	3	1	15	15	7	15	15	7
JAESCHKE-3	9	9	7	5	4	9	5	4	25	5	4
LUTZ1-3	32	32	43	13	829	159	31	1263	3741	31	1263
MANSOOR-3	11	11	7	1	0	19	19	53	19	19	53
MITCHELL-3	21	21	19	7	5	129	23	7	129	23	7
ROSZIEG-3	25	25	17	9	13	191	25	16	191	25	16
SAWYER30-3	30	30	59	25	4	229	57	37	1035	129	37
AKKSKT12-3	12	12	11	11	62	19	11	62	19	11	62
AKO20T-3	15	15	3	1	0	9	1	0	225	41	3
AKO30T1-3	16	16	25	23	7	27	21	19	27	21	19
AKO30T2-3	14	14	3	3	8	11	11	12	11	11	12
GUN17T-3	17	17	15	15	58	35	15	58	67	33	58
GUN8T-3	8	8	3	3	5	5	5	7	13	11	9
LAM20T-3	9	9	3	3	38	5	3	38	5	3	38
LAM30T-3	9	9	5	5	8	9	5	8	13	11	9
MAS30T-3	30	30	37	25	191	419	61	191	5843	337	191
MGG7T-3	7	7	7	5	6	7	7	11	7	7	11
WANG18T-3	20	20	33	17	14	179	21	17	179	21	17
YKA19T-3	19	19	27	17	6	53	19	6	53	19	6
YKA27T-3	27	27	21	17	19	21	17	19	21	17	19
YKA31T-3	31	31	97	29	11	173	35	13	173	35	13

Table 6-18 B&B results for Problem I make span analysis

	PROBLEM I				PROBLEM II						
Test Problem	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	Average Reduction in Number of Nodes	Average Reduction in CPU Times	Average Deviation of Optimal Solution Values
BOWMAN8-3	17.4	11.6	0.011	11.9	9.8	8.4	0.008	10.2	43.7	29.1	12.5
BUXEY-3	2982.6	223.2	5.072	23.1	331.4	29.9	0.605	13.5	88.9	88.1	43.3
MERTENS-3	20.4	7.2	0.011	2.4	7.4	4.0	0.008	1.2	63.7	28.4	40.8
HESKIA-3	222.2	41.8	0.261	310.1	29.2	29.2	0.023	310.1	86.9	91.0	0.0
JACKSON-3	20.8	14.8	0.014	6.5	15.4	14.0	0.011	6.3	26.0	22.7	8.3
JAESCHKE-3	23.4	8.4	0.011	3.7	4.8	3.3	0.008	1.1	79.5	28.4	78.3
LUTZ1-3	20184.6	34.0	38.315	1230.6	262.2	13.4	0.397	821.1	98.7	99.0	32.8
MANSOOR-3	17.0	17.0	0.014	51.8	14.4	14.2	0.009	39.1	15.3	33.3	24.3
MITCHELL-3	1129.2	55.8	0.907	15	39.0	18.6	0.022	14.1	96.5	97.6	5.7
ROSZIEG-3	6286.0	24.0	7.520	14	91.4	15.2	0.064	12.4	98.5	99.1	10.9
SAWYER30-3	2219.2	168.8	3.760	29.1	338.8	36.0	0.627	28.8	84.7	83.3	1.0
AKKSKT12-3	15.4	10.8	0.012	64.7	11.0	9.8	0.008	63.9	28.6	36.8	1.1
AKO20T-3	113.8	39.6	0.039	13.5	41.8	16.5	0.023	10.8	63.3	40.2	24.9
AKO30T1-3	118.0	36.4	0.058	21.2	32.6	22.4	0.017	21.2	72.4	70.2	0.0
AKO30T2-3	49.4	31.6	0.023	19	16.4	13.6	0.011	18.4	66.8	53.4	2.6
GUN17T-3	66.4	23.6	0.036	64.4	24.4	17.4	0.014	64.4	63.3	61.1	0.0
GUN8T-3	38.8	13.4	0.012	10.5	3.8	3.8	0.006	6.1	90.2	49.6	44.4
LAM20T-3	7.2	6.0	0.012	65.5	4.2	4.2	0.006	60.7	41.7	49.6	7.1
LAM30T-3	16.4	10.6	0.012	7.9	5.8	5.2	0.008	4.5	64.6	37.6	42.8
MAS30T-3	8021.0	224.0	11.654	194.5	101.0	29.4	0.089	194.5	98.7	99.2	0.0
MGG7T-3	10.0	6.0	0.011	5.8	5.6	4.4	0.006	4.6	44.0	43.6	15.5
WANG18T-3	136.2	17.6	0.077	29.1	34.0	17.6	0.027	29.1	75.0	65.4	0.0
YKA19T-3	49.8	12.8	0.028	9.2	23.4	13.2	0.020	8.2	53.0	27.8	9.4
YKA27T-3	32.2	25.6	0.031	21.1	31.4	25.4	0.036	20.8	2.5	-15.0	1.6
YKA31T-3	397.8	27.2	0.536	11.9	109.6	25.2	0.205	11.3	72.4	61.8	5.2

6.5 LARGE SIZE PROBLEMS

Our large problem set is comprised of 10 problems which are also obtained from the ALBP data base. The characteristic of those problems are presented in Table 6-19. By using three different *m/n* ratios and three cycle times we solved 90 problem instances.

Table 6-19 The characteristics of the large problems

Problem	Number of Tasks	Number of Precedence Relations	Number of Parts		
			$\frac{m}{n} = 50\%$	$\frac{m}{n} = 75\%$	$\frac{m}{n} = 100\%$
GUNTHER	35	45	18	27	35
KILBRID	45	62	23	34	45
HAHN	53	82	27	40	53
WARNECKE	58	70	29	44	58
TONGE70	70	86	35	53	70
WEE-MAG	75	38	38	57	75
ARC83	83	42	42	63	83
LUTZ2	89	45	45	67	89
LUTZ3	89	45	45	67	89
MUKHERJE	94	47	47	71	94

Table 6-20 shows the results obtained for UB2 and min {UB1, UB2, UB3} for a fixed maximum number of nodes (Table 6-20 contains the results for cycle time level 50% and *m/n* ratio 75%, complete results including the remaining factor levels are presented in Appendix C). Form Table 6-20 we can observe that for UB2 and UB_BEST even though there are differences between the total numbers of B&B nodes, the optimality node indices are mostly the same. The objective function value obtained by UB2 and UB_BEST are the same for all the cases. For all the problems in Table 6-20, UB2 outperforms UB_BEST in CPU time and the difference between UB2 and UB_BEST becomes more significant as *m* increases.

For the whole test set, UB2 performed better than UB_Best in 76% of the cases. Hence, we verify that our observations made on the upper bounds for the small sized problems are also valid for our large sized problems.

Table 6-20 B&B results for large sized problems (with fixed node numbers) (*CT*=%50, *m/n*=75%)

			UB2					UB_BEST				
Test Problem	***n***	***m***	**# of Nodes**	**Optimal Node Index**	**CPU Time**	**Max CPU Time**	**Optimal Solution Value**	**# of Nodes**	**Optimal Node Index**	**CPU Time**	**Max CPU Time**	**Optimal Solution Value**
GUNTHER-2-0	35	27	1055	25	0.6294	0.6410	70	1029	25	0.7825	0.7970	70
KILBRID-2-0	45	34	1449	47	2.4837	2.5140	119	1447	45	3.0508	3.0800	119
HAHN-2-0	53	40	531	55	0.8557	0.9020	1668	391	55	0.8211	0.8530	1668
WARNECKE-2-0	58	44	7113	57	14.5575	14.7890	74	6941	57	17.0335	17.2440	74
TONGE70-2-0	70	53	10001	41	21.9420	22.0010	632	10001	41	25.9185	25.9780	632
WEE-MAG-2-0	75	57	10001	383	96.8630	97.4900	219	10001	381	104.2346	104.8910	219
ARC83-2-0	83	63	10001	101	67.1244	67.3320	8156	10001	101	74.0077	74.2250	8156
LUTZ2-2-0	89	67	10001	303	82.1478	82.2950	26	10001	303	89.0364	89.2100	26
LUTZ3-2-0	89	67	10001	129	72.2211	77.1500	206	10001	129	78.0026	81.3560	206
MUKHERJE-2-0	94	71	10001	273	168.9906	170.5270	576	10001	249	179.2536	182.3550	576

To analyze the efficiency of the algorithm for large problem instances, we will generate results by using UB2 under one hour time limit. The results obtained are presented in Table 6-24, 6-25 and 6-26.

For 25 cases we reached the time limit of 1 hour. For 9 of those cases, we observed deviation from the optimal solutions objective function value. Deviations are observed for problems having part numbers greater than 70. The average deviation is less than 8%, with a maximum of 19.2% observed for WEE-MAG. The deviation from the optimal solution value is increasing with the increasing number of parts as expected. Even though the optimal node to number of nodes ratio is decreasing for large problems for some of the instances, the algorithm is not efficient for solving problems within short time limits.

To discuss the effects of the cycle time and *m/n* ratio on total number of B&B nodes, CPU time and optimal node index ratio we present Table 6-21, 6-19 and 6-20 respectively.

From the tables one can observe that the behaviors (tendencies of increase and decrease) for large size problems are similar to our small sized problems. From the Figures 6.7 and 6.8 one can see that the graphics are different. This difference is due to the run time limitation we have used for large problem instances.

Table 6-21 Number of B&B nodes with changing cycle time and *m/n* ratio (Large Problems)

		Cycle Time as a % of Total Task Times					
		25%		**50%**		**75%**	
		Average	**Max**	**Average**	**Max**	**Average**	**Max**
m/n	**0.50**	71.4	181	9340.4	41695	54433	239125
	0.75	2487.8	22575	80079.8	282063	103115	262457
	1.00	19240	181871	120814	407187	139751	338043

Table 6-22 CPU time with changing cycle time and *m/n* ratio (Large Problems)

		Cycle Time as a % of Total Task Times					
		25%		50%		75%	
		Average	Max	Average	Max	Average	Max
m/n	**0.50**	0.29820006	0.734	57.6283	251.625	451.988	2155.44
	0.75	24.9609	240.875	1361.44	3600.02	1882.97	3600.02
	1.00	366.0497	3600.01	1969.04	3600.11	2529.24	3600.13

Table 6-23 Optimal node index ratio with changing cycle time and *m/n* ratio (Large Problems)

		Cycle Time as a % of Total Task Times		
		25%	50%	75%
		Average	Average	Average
m/n	**0.50**	0.57293925	0.16303	0.13789
	0.75	0.23984754	0.02188	0.00333
	1.00	0.20880161	0.00592	0.00551

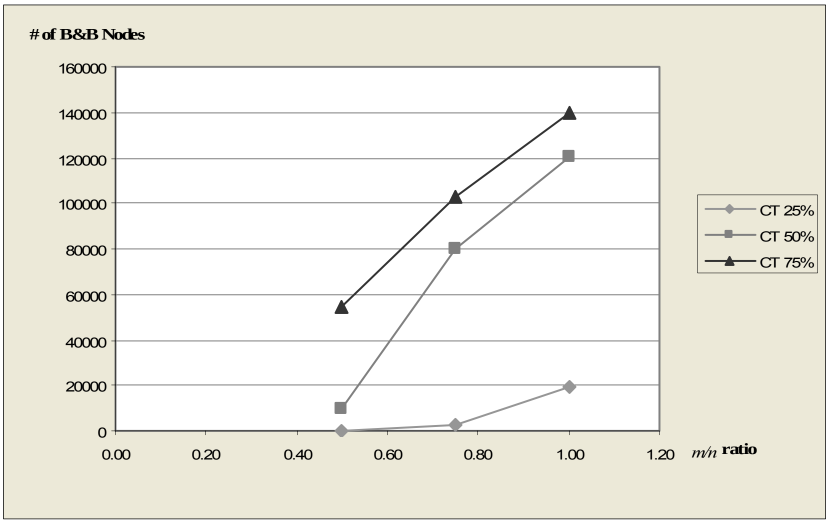


Figure 6.7 Number of B&B nodes versus *m/n* ratio (Large Problems)

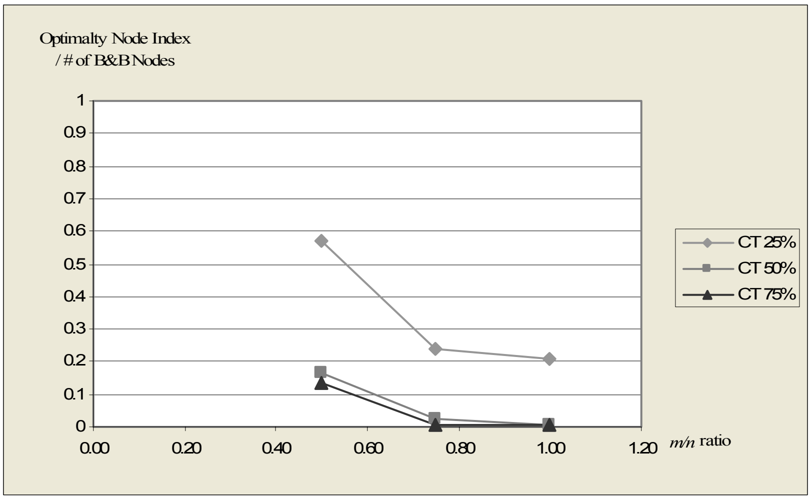


Figure 6.8 Optimality node index ratio versus *m/n* ratio (Large Problems)

6.6 SUMMARY OF THE RESULTS

A summary of the obtained results are as follows;

- UB2 is the best upper bound for all the problem sizes considered.
- Reduction Mechanisms are effective in reducing the size of the B&B tree and the necessary CPU time. The percentage reduction in the number of nodes and the CPU times increases with an increase in cycle times.
- The ratio of the optimal node index to the number of nodes in the B&B tree decreases as the problem size increases.
- Optimal solution values obtained for 50% cycle time level and 75% cycle time level is generally the same.
- For small cycle time levels, large problem sets can also be solved to optimality with B&B algorithm.
- For large problems optimal solution is found in 87% of the problem instances. The average deviation is 8% and the maximum deviation is 19.2%

.

Table 6-24 B&B results for large sized problems (with limited run time) (CT=%25)

Test Problem	*n*	*m*	# of Nodes	Opt. Node Index	CPU Time	Solution Value	Optimal Solution Value	Percent Deviation From Opt. Sol.
GUNTHER-1-0	35	18	13	13	0.0150	42	42	0.00
GUNTHER-2-0	35	27	19	13	0.0000	28	28	0.00
GUNTHER-3-0	35	35	75	21	0.0310	51	51	0.00
KILBRID-1-0	45	23	67	27	0.0470	151	151	0.00
KILBRID-2-0	45	34	203	35	0.2030	119	119	0.00
KILBRID-3-0	45	45	55	39	0.0940	105	105	0.00
HAHN-1-0	53	27	11	11	0.0930	530	530	0.00
HAHN-2-0	53	40	5	0	0.1410	0	0	0.00
HAHN-3-0	53	53	31	21	0.1570	57	57	0.00
WARNECKE-1-0	58	29	25	11	0.0160	142	142	0.00
WARNECKE-2-0	58	44	151	35	0.1250	59	59	0.00
WARNECKE-3-0	58	58	601	43	0.6250	81	81	0.00
TONGE70-1-0	70	35	47	25	0.0620	716	716	0.00
TONGE70-2-0	70	53	81	41	0.1720	632	632	0.00
TONGE70-3-0	70	70	581	63	1.7970	655	655	0.00
WEE-MAG-1-0	75	38	91	59	0.6090	393	393	0.00
WEE-MAG-2-0	75	57	22575	81	240.8750	215	215	0.00
WEE-MAG-3-0	75	75	181871	3275	3600.0140	120	120	0.00
ARC83-1-0	83	42	181	41	0.3440	8966	8966	0.00
ARC83-2-0	83	63	1121	75	3.7030	3510	3510	0.00
ARC83-3-0	83	83	5759	77	37.3570	3107	3107	0.00
LUTZ2-1-0	89	45	69	45	0.7030	13	13	0.00
LUTZ2-2-0	89	67	111	0	1.2810	0	0	0.00
LUTZ2-3-0	89	89	385	41	1.9840	9	9	0.00
LUTZ3-1-0	89	45	81	35	0.7340	126	126	0.00
LUTZ3-2-0	89	67	135	49	1.5310	117	117	0.00
LUTZ3-3-0	89	89	1271	65	7.7970	138	138	0.00
MUKHERJE-1-0	94	47	129	51	0.3590	627	627	0.00
MUKHERJE-2-0	94	71	477	77	1.5780	361	361	0.00
MUKHERJE-3-0	94	94	1771	93	10.6410	412	412	0.00

Table 6-25 B&B results for large sized problems (with limited run time) (*CT*=%50)

Test Problem	*n*	*m*	# of Nodes	Opt. Node Index	CPU Time	Solution Value	Optimal Solution Value	Percent Deviation From Opt. Sol.
GUNTHER-1-0	35	18	299	27	0.1880	81	81	0.00
GUNTHER-2-0	35	27	1055	25	1.1250	70	70	0.00
GUNTHER-3-0	35	35	6901	35	18.6570	68	68	0.00
KILBRID-1-0	45	23	67	27	0.0780	151	151	0.00
KILBRID-2-0	45	34	1449	47	4.5470	119	119	0.00
KILBRID-3-0	45	45	8593	89	56.5470	105	105	0.00
HAHN-1-0	53	27	237	39	0.3910	2319	2319	0.00
HAHN-2-0	53	40	531	55	1.5630	1668	1668	0.00
HAHN-3-0	53	53	12367	75	68.2970	2054	2054	0.00
WARNECKE-1-0	58	29	435	11	0.7650	142	142	0.00
WARNECKE-2-0	58	44	7113	57	26.3910	74	74	0.00
WARNECKE-3-0	58	58	407187	77	3286.7240	85	85	0.00
TONGE70-1-0	70	35	1335	55	2.2190	716	716	0.00
TONGE70-2-0	70	53	14483	41	57.8750	632	632	0.00
TONGE70-3-0	70	70	180505	561	1521.9840	655	655	0.00
WEE-MAG-1-0	75	38	241	73	1.8130	393	393	0.00
WEE-MAG-2-0	75	57	226115	383	3600.0160	219	219	0.00
WEE-MAG-3-0	75	75	116213	96541	3600.0630	118	120	1.67
ARC83-1-0	83	42	1225	63	5.6400	14870	14870	0.00
ARC83-2-0	83	63	282063	101	3600.0010	8156	8156	0.00
ARC83-3-0	83	83	128987	123	3600.0270	8202	8202	0.00
LUTZ2-1-0	89	45	37935	20579	222.9380	50	50	0.00
LUTZ2-2-0	89	67	236191	303	3600.0120	26	28	7.14
LUTZ2-3-0	89	89	126751	14429	3600.0160	17	19	10.53
LUTZ3-1-0	89	45	41695	57	251.6250	268	268	0.00
LUTZ3-2-0	89	67	271987	129	3600.0160	206	206	0.00
LUTZ3-3-0	89	89	160545	91	3600.0150	194	194	0.00
MUKHERJE-1-0	94	47	9935	77	90.6260	975	975	0.00
MUKHERJE-2-0	94	71	117905	273	3600.0150	576	576	0.00
MUKHERJE-3-0	94	94	60093	1263	3600.1080	532	532	0.00

Table 6-26 B&B results for large sized problems (with limited run time) (*CT*=%75)

Test Problem	*n*	*m*	# of Nodes	Opt. Node Index	CPU Time	Solution Value	Optimal Solution Value	Percent Deviation From Opt. Sol.
GUNTHER-1-0	35	18	611	27	0.6560	81	81	0.00
GUNTHER-2-0	35	27	2881	39	6.2970	84	84	0.00
GUNTHER-3-0	35	35	21777	61	88.6710	74	74	0.00
KILBRID-1-0	45	23	67	27	0.1410	151	151	0.00
KILBRID-2-0	45	34	18449	35	117.9060	119	119	0.00
KILBRID-3-0	45	45	338043	2337	3600.0160	105	105	0.00
HAHN-1-0	53	27	723	39	1.5620	2319	2319	0.00
HAHN-2-0	53	40	7473	55	33.4210	1668	1668	0.00
HAHN-3-0	53	53	80961	75	686.6870	2054	2054	0.00
WARNECKE-1-0	58	29	995	11	2.4070	142	142	0.00
WARNECKE-2-0	58	44	83393	59	506.0710	74	74	0.00
WARNECKE-3-0	58	58	252061	81	3600.0120	93	93	0.00
TONGE70-1-0	70	35	98743	713	518.1090	716	716	0.00
TONGE70-2-0	70	53	262457	41	3600.0160	632	632	0.00
TONGE70-3-0	70	70	105037	43	3600.0160	648	655	1.07
WEE-MAG-1-0	75	38	241	73	1.9060	393	393	0.00
WEE-MAG-2-0	75	57	160057	377	3600.0210	219	219	0.00
WEE-MAG-3-0	75	75	61269	30473	3600.1260	97	120	19.17
ARC83-1-0	83	42	102567	65	927.3010	14870	14870	0.00
ARC83-2-0	83	63	170493	123	3600.0160	8091	8156	0.80
ARC83-3-0	83	83	75179	153	3600.0280	7643	8202	6.82
LUTZ2-1-0	89	45	91323	50033	821.7230	50	50	0.00
LUTZ2-2-0	89	67	195153	303	3600.0040	26	28	7.14
LUTZ2-3-0	89	89	88659	63	3600.0470	16	19	15.79
LUTZ3-1-0	89	45	239125	91	2155.4380	268	268	0.00
LUTZ3-2-0	89	67	172281	407	3600.0060	206	206	0.00
LUTZ3-3-0	89	89	85571	91	3600.0460	194	194	0.00
MUKHERJE-1-0	94	47	9935	77	90.6420	975	975	0.00
MUKHERJE-2-0	94	71	117929	273	3600.0150	576	576	0.00
MUKHERJE-3-0	94	94	60093	1263	3600.0150	532	532	0.00

CHAPTER 7

CONCLUSION AND FUTURE RESEARCH

The disassembly of a product, in particular at the end of its useful life, is becoming a common and worthwhile industrial practice due to government legislations and the economical benefits.

Disassembly processes in their simplest sense are comprised of disassembly activities, their precedence relations and the parts released. Disassembly is employed for recovery of valuable and reusable parts or subassemblies, product separation to facilitate the downstream material recovery process, the removal of hazardous or toxic materials, to remanufacturing of the product for another useful life, and getting rid of the proprietary parts or subassemblies. Depending on those reasons the extent of the disassembly process must be determined by keeping environmental, economical and technical issues in mind.

In this study, we considered the disassembly of products for recovering valuable parts. All parts obtained by disassembly processes of an EOL product may not be profitable due to their high recovery costs. Our problem, in its simplest form, is to select the parts to be released and determine the associated disassembly activities so as to maximize the total profit.

We handled two versions of the problem: the simple part selection problem and the part selection problem with cycle time constraint.

We provided Mixed Integer Problem formulations of both versions of the problem. We showed that the constraint set of the first problem is totally unimodular yielding integral solutions. We also provided the dual formulation of the simple part selection problem together with its interpretation.

We developed a reduction mechanism to reduce the size of the both problems. The reduction mechanism is generated through detailed analysis of the precedence graphs and problem parameters such as costs, times and revenues and effectively reduces problem sizes.

For our second problem, we proposed a branch-and-bound algorithm incorporating several elimination and bounding mechanisms. We found that those mechanisms are quite efficient in reducing the size of the search.

We conducted an experimental study to test the performance of the branch and bound algorithm and the reduction algorithm. We showed that our algorithm is capable of solving large size problem instances having up to 94 tasks and parts.

Our results can be extended to several research areas most noteworthy of which are listed below:

- Reduction mechanisms and branch and bound can be extended to include OR type relations frequently encountered in disassembly processes. Such an inclusion may necessitate the design of a task-based branch and bound algorithm.
- Disassembly operations require different tools to be used and the change of product orientation so the sequence dependent task times and costs are mostly present in disassembly operations. Thus sequence dependency can be considered for further research. Yet when only AND type relations are considered sequence dependency is a simple task. If OR type relations are introduced sequence dependency considerations will be more important. The inclusion of sequence dependency into the branch and bound algorithm is rather simple if a task based algorithm is developed. But the reduction algorithm needs to be re-evaluated.

- Disassembly line balancing problem having fixed or variable number of stations may be worth-studying. Our approach is a special case of profit oriented selective disassembly line balancing problem with a single workstation. Reduction mechanisms developed in this study can be extended to disassembly line balancing problems.
- Our extensive computational study has revealed that our approach is capable of solving instances with up to 94 parts. Heuristic approaches for even larger sized problems, might be worth developing.

REFERENCES

Altekin , F.T, Profit oriented disassembly line balancing, Ph.D. Thesis, METU, Department of Industrial Engineering, 2005.

Chung, C., Peng, Q., An integrated approach to selective-disassembly sequence planning, Robotics and Computer-Integrated Manufacturing 2005; 21: 475-485.

Das, S. K., Caudill, R., The design of high volume disassembly lines, Demanufacturing of Electronic Equipment for Reuse and Recycling information Exchange Meeting Archives, 25-26 October 1999.

Das, S. K., Yedlarajiah, P., Narendra, R., An approach for estimating the end-of-life product disassembly effort and cost, International Journal of Production Research 2000; 38: 657- 673.

De Ron, A., Penev, K., Disassembly and recycling of electronic consumer products: an overview, Technovation 1995; 15: 363-374.

Dutta, P., Majumder, D.D., Performance evaluation of evolutionary class of algorithms-An application to 0–1 knapsack problem, Mathematical and Computer Modeling 1998; 27: 57-72.

Fleischmann M., Bloemhof-Ruwaard J. M., Dekker R., Van Der Laan E., Van Nunen J.A.E.E., Van Wassenhove L. N., Quantitative models for reverse logistics: A review, European Journal of Operational Research 1997; 103: 1-17.

Gao, M., Zhou, M.C., Caudill, R.J., Integration of disassembly leveling and bin assignment for demanufacturing automation, IEEE Transactions on Robotics and

Automation 2002; 18: 867-874.

Gonzalez, B., Adenso-Diaz, A bill of materials-based approach for end-of-life decision making in design for environment, International Journal of Production Research 2005; 43: 2071-2099.

Gonzalez, B., Adenso-Diaz, B., A scatter search approach to the optimum disassembly sequence problem, Computers & Operations Research 2006; 33: 1776-1793.

Gupta, S.M., Taleb, K.N., Scheduling disassembly, International Journal of Production Research 1994; 32: 1857- 1866.

Güngör, A., Gupta, S.M., An evaluation methodology for disassembly processes, Computers & Industrial Engineering 1997; 33: 329-332.

Güngör, A., Gupta, S.M., Issues in environmentally conscious manufacturing and product recovery: a survey, Computer & Industrial Engineering 1999; 36: 811-853.

Güngör, A., Gupta, S.M., Disassembly sequence plan generation using a branch-and-bound algorithm, International Journal of Production Research 2001(a); 39: 481-509.

Güngör, A., Gupta, S.M., A solution approach to the disassembly line balancing problem in the presence of task failures, International Journal of Production Research 2001(b); 39: 1427-1467.

Güngör, A., Gupta, S.M., Disassembly line in product recovery, International Journal of Production Research 2002; 40: 2569-2589.

Homem de Mello, L.S., Sanders A., AND/OR graph representation of assembly plans, IEEE Transactions on Robotics and Automation 1990; 6: 188-199.

Hula, A., Jalali, K., Hamza, K., Skerlos, S.J., Saitou, K., Multi-criteria decision-making for optimization of product disassembly under multiple situations, Environmental Science & Technology 2003; 37: 5303-5313.

Johnson, M.R., Wang, M.H., Economical evaluation of disassembly operations for recycling, remanufacturing and reuse, International Journal of Production Research 1998; 36(12): 3227-3252.

Lambert, A.J.D., Optimal disassembly of complex products, International Journal of Production Research 1997; 35: 2509-2523.

Lambert, A.J.D., Linear programming in disassembly/clustering sequence generation, Computer & Industrial Engineering 1999; 36: 723-738.

Lambert, A.J.D., Determining optimum disassembly sequences in electronic equipment, Computer & Industrial Engineering 2002; 43: 553-575.

Lambert, A.J.D., Disassembly sequencing: a survey, International Journal of Production Research 2003; 41: 3721-3759.

Lambert, A.J.D., Exact methods in optimum disassembly sequence search for problems subject to sequence dependent costs, The International Journal of Management Science 2006a (forthcoming).

Lambert, A.J.D., Optimizing disassembly processes subjected to sequence-dependent cost, Computers & Operations Research 2006b (forthcoming).

Lambert, A.J.D., Gupta, S.M., Demand-driven disassembly optimization for electronic products, Journal of Electronics Manufacturing 2002; 11: 121-135.

Lee, D-H., Kang, J-G., Xirouchakis, P., Disassembly planning and scheduling: review and further research, Proceedings of the Institution of Mechanical Engineers 2001; 215: 695-709.

Li, J.R., Khoo, L.P., Tor, S.B., An object oriented intelligent disassembly sequence planner for maintenance, Computers in Industry 2005; 56: 699-718.

Li, J.R., Khoo, L.P., Tor, S.B., Generation of possible multiple components disassembly sequence for maintenance using a disassembly constraint graph, International Journal of Production Economics 2006 (forthcoming).

Mascle, C., Balasoiu B-A., Algorithmic selection of a disassembly sequence of a component by a wave propagation method, Robotics and Computer Integrated Manufacturing 2003; 19: 439-448.

Moore, K.E., Güngörö A., Gupta S.M., A Petri-net approach to disassembly process planning, Computers & Industrial Engineering 1998; 35: 165-168.

Moyer, L., Gupta, S.M., Environmental concerns and recycling/disassembly efforts in the electronics industry, Journal of Electronics Manufacturing 1997; 7: 1-22.

Navin-Chandra, D., The recovery problem in product design, Journal of Engineering Design 1994; 5:65-86.

O'shea, B., Grewal, S.S. and Kaebernick, H., State of the art literature survey on disassembly planning, Concurrent Engineering 1998; 6: 345–357.

Scholl, A., Balancing and sequencing of assembly lines, Physica-Verlag

Heidelberg 1999; Second edition.

Schrijver, A., Theory of Linear and Integer Programming, Wiley 1998.

Tang, Y., Zhou, M.C., Zussman, E. and Caudill, R.,Disassembly modeling, planning, and application: a review, Proceedings of IEEE International Conference on Robotics and Automation 2000; 3: 2197–2202.

Thierry, M., Salomon, M., Nunen, J. V., Wassenhove, L. V., Strategic issues in product recovery management, California Management Review 1995; 37: 114-135.

Toffel, M.W., End-of-life product recovery: drivers, prior research, and future directions, Conference on European Electronics Take-back Legislation: Impacts on Business Strategy and Global Trade, Fontainebleau 2002.

APPENDIX A

AN ILLUSTRATIVE EXAMPLE FOR BRANCH AND BOUND ALGORITHM

As an illustrative example we will use the precedence graph and task properties of JAESCHKE from the ALBP set. We use UB2 to evaluate the partial solutions. The revenues are generated as defined in Chapter 6. There are 9 tasks and 7 parts available in the problem; their associated parameters are given in Table A-1 and the precedence graph of the problem is in Table B-1. Cycle time is equal to the half of the total task times (i.e., $CT = \left\lceil \frac{\sum t_i}{2} \right\rceil = \frac{6 + ... + 15}{2} = 19$)

Table A-1 Task costs, task times and part revenues of the example problem

i	Task Costs (c_i)	Task Times (t_i)	j	Part revenues (p_j)
1	6	6	**1**	4
2	4	4	**2**	4
3	1	1	**3**	6
4	5	5	**4**	5
5	4	4	**5**	6
6	5	5	**6**	3
7	4	4	**7**	7
8	3	3		
9	5	5		

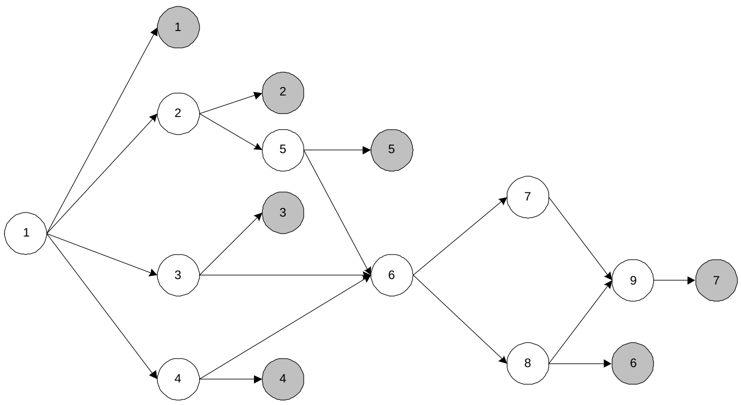


Figure A.1 Precedence graph of the example problem
(The gray nodes in the graph corresponds to parts)

Solution Scheme:

Step 1. // Input //

- **Input the DPM, the PN, part revenues, task costs and task times.**

$$R = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad PN = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \\ 6 & 8 \\ 7 & 9 \end{bmatrix}$$

Step 2. // Initial Reduction //

- Create PT using DPM and PN

$$PT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Note that S_j can be created using the j^{th} row of PT and $\underline{U_i}$ can be created using the i^{th} column of PT

- **Apply Part Time Control (Reduction Rule 2) using PT and task times.**

Task times of all the available parts are as follows:

j	1	2	3	4	5	6	7
$\sum_{i \in S_j} t_i$	6	10	7	11	14	28	37

As cycle time is 19, part 6 and part 7 cannot be recovered so the reduced parts and tasks sets are:

$$P = \{1,2,3,4,5\}$$

$$T = \{1,2,3,4,5\}$$

- **Apply Part and Task Reduction (Reduction Rule 3) using PT, part revenues and task costs.**

i	1	2	3	4	5
$\sum_{j \in U_i} p_j - c_i$	19	6	5	0	1

No actions performed as all $\sum_{j \in U_i} p_j - c_i \geq 0$

- **Apply Task Aggregation (Reduction Rule 5) using DPM and PN.**

No task aggregation is possible as all tasks are releasing parts.

- **Apply Part Aggregation (Reduction Rule 4) using PT and part revenues.**

No part aggregation is possible as all tasks are releasing only one part.

- **Create the reduced problem with new input matrices (PT, part revenues, task costs and task times.**

$$PT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{aligned} c_i &= [6 \quad 4 \quad 1 \quad 5 \quad 4] \\ t_i &= [6 \quad 4 \quad 1 \quad 5 \quad 4] \\ p_j &= [4 \quad 4 \quad 6 \quad 5 \quad 6] \end{aligned}$$

The reduced precedence graph is:

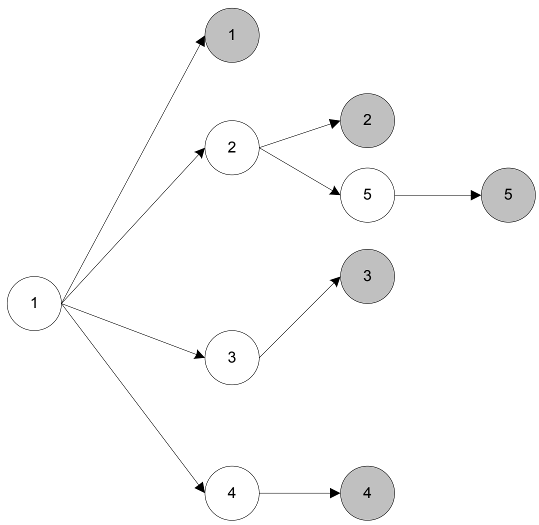


Figure A.2 Reduced precedence graph of the problem

Step 3. //Formation//

- **Calculate the initial lower bound and incumbent solution accordingly.**

j	1	2	3	4	5
$\max\left(p_j - \sum_{i \in S_j} c_i \ , \ 0\right)$	0	0	0	0	0

LB=0

$z_{inc} = 0$

P_{inc}=Ø

- **Form the initial problem assuming all $x_j = 0$.**

  Problem No = 1

  $P = \{1,2,3,4,5\}$ P^*=Ø P'=Ø

  $T = \{1,2,3,4,5\}$ T^*=Ø

Step 4. // Reduction//

- **Calculate the remaining cycle time.**

  CT=19

- **Collect the parts with all their tasks performed.**

  No actions taken

- **Apply Part Time Control (Reduction Rule 2) using PT and task times.**

  No actions taken

- **Apply Part and Task Reduction (Reduction Rule 3) using PT, part revenues and task costs.**

  No actions taken

- **Apply Part Aggregation (Reduction Rule 4) using PT and part revenues.**

  No actions taken

- **Apply Task Insertion into Parts (Reduction Rule 6).**

  As $|U_3|$, $|U_4|$ and $|U_5|$ are equal to 1, the associate part revenues will be altered as follows:

j	3	4	5
p_j	5	0	2
tp_j	1	5	4

$$PT = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad \begin{aligned} c_i &= [6 \quad 4] \\ t_i &= [6 \quad 4] \\ p_j &= [4 \quad 4 \quad 5 \quad 0 \quad 2] \\ tp_j &= [0 \quad 0 \quad 1 \quad 5 \quad 4] \end{aligned}$$

Step 8. // Sub-problem //

- **Calculate the objective function, i.e., profit value of the sub-problem.**

  z = 0

- **Calculate the upper bound value of the sub-problem using UB2.**

$$UB = \sum_{i \in T} \max\left(\sum_{j \in U_i} \frac{p_j}{|S_j|} - c_i,\ 0 \right)$$

i	1	2	3	4	5
$\sum_{j \in U_i} \frac{p_j}{\lvert S_j \rvert} - c_i$	6	0	0	0	0

  UB=6

- **If the lower bound is less than the sub-problem's objective function value then update the incumbent solution as the sub-problem and the lower bound value.**

  $LB \geq z$ **,** No actions taken

- **If the upper bound value of the sub-problem is less than the lower bound then prune the branch and go to Step 7.**

  $UB > LB$ so we do not prune the branch.

- **If a decision for all the parts are made then go to Step 7.**

  As $|P| > 0$, termination condition is not satisfied.

Step 9. **// Branching //**

- **Select the part j to be branched according to equation (5.3).**

j	1	2	3	4	5
$\max\left(\dfrac{p_j - \sum_{i\in S_j}\dfrac{c_i}{\lvert U_i\rvert}}{\sum_{i\in S_j} t_i + tp_j},\ 0\right)$	0.47	0	0.54	0	0

Part 3 is selected for branching.

- **Create two new sub-problems.**

Problem No = 2

$P = \{1,2,4,5\}$ $P^*=\{3\}$ $P'=\emptyset$

$T = \{2\}$ $T^*=\{1\}$

Problem No = 3

$P=\{1,2,4,5\}$ $P^*= \emptyset$ $P'=\{3\}$

$T=\{1,2\}$ $T^*= \emptyset$

Step 10. **// Node Selection //**

- **Select the sub-problem to be solved using the depth first search strategy and go to Step 4.**

Problem 2 will be solved next

Step 11. **// Reduction//**

- **Calculate the remaining cycle time.**

CT=12

- **Collect the parts with all their tasks performed.**

The recovery of part 3 imposed the realization of task 1 , therefore part 1 is automatically released.

So :

$P = \{2,4,5\}$ $P^*=\{2,3\}$ $P'=\emptyset$

$T = \{2\}$ $T^{*}=\{1\}$

- **Apply Part Time Control (Reduction Rule 2) using PT and task times.**

  No actions taken

- **Apply Part and Task Reduction (Reduction Rule 3) using PT, part revenues and task costs.**

  No actions taken

- **Apply Part Aggregation (Reduction Rule 4) using PT and part revenues.**

  No actions taken

- **Apply Task Insertion into Parts (Reduction Rule 6).**

  No actions taken

Step 12. // Sub-problem //

- **Calculate the objective function, i.e., profit value of the sub-problem.**

  z = 3

- **Calculate the upper bound value of the sub-problem using UB2.**

$$UB = \sum_{i \in T} \max\left(\sum_{j \in U_i} \frac{p_j}{|S_j|} - c_i,\ 0 \right)$$

i	2	4	5
$\sum_{j \in U_i} \frac{p_j}{\lvert S_j \rvert} - c_i$	2	0	0

UB=2

As $z > 0 \rightarrow UB = UB + z$

$UB=5$

- **If the lower bound is less than the sub-problem's objective function value then update the incumbent solution as the sub-**

problem and the lower bound value.

LB=3

$z_{inc} = 3$

$P_{inc}=\{1,3\}$

- **If the upper bound value of the sub-problem is less than the lower bound then prune the branch and go to Step 7.**

  $UB > z_{inc}$ so we do not prune the branch.

- **If a decision for all the parts are made then go to Step 7.**

  As $|P| > 0$ termination condition is not satisfied.

Step 13. // Branching //

- **Select the part j to be branched according to equation (5.3).**

j	2	4	5
$\max\left(\dfrac{p_j - \sum_{i\in S_j}\frac{c_i}{\lvert U_i\rvert}}{\sum_{i\in S_j} t_i + tp_j},\ 0\right)$	0.5	0	0

Part 2 is selected for branching.

- **Create two new sub-problems.**

  Problem No = 4

  $P = \{4,5\}$ $P^{*}=\{1,2,3\}$ $P'=\emptyset$

  $T=\emptyset$ $T^{*}=\{1,2\}$

  Problem No = 5

  $P = \{4,5\}$ $P^{*}=\{1,3\}$ $P'=\{2\}$

  $T = \{2\}$ $T^{*}=\{1\}$

We will stop the procedure here; optimal solution of the original problem is recovery of parts 1, 2, 3 and 5 with the realization of tasks 1, 2, 3 and 5. Rest of the solution scheme can be seen in Figure A.4. An illustrative example on how the nodes are defined is presented in Figure A.3.

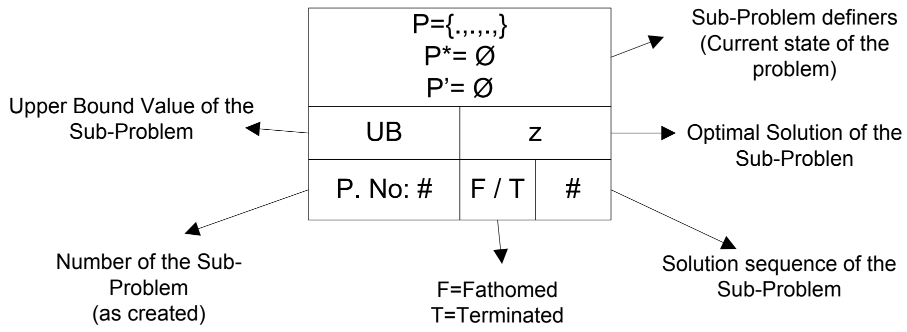


Figure A.3 Example on Sub-Problem node definitions

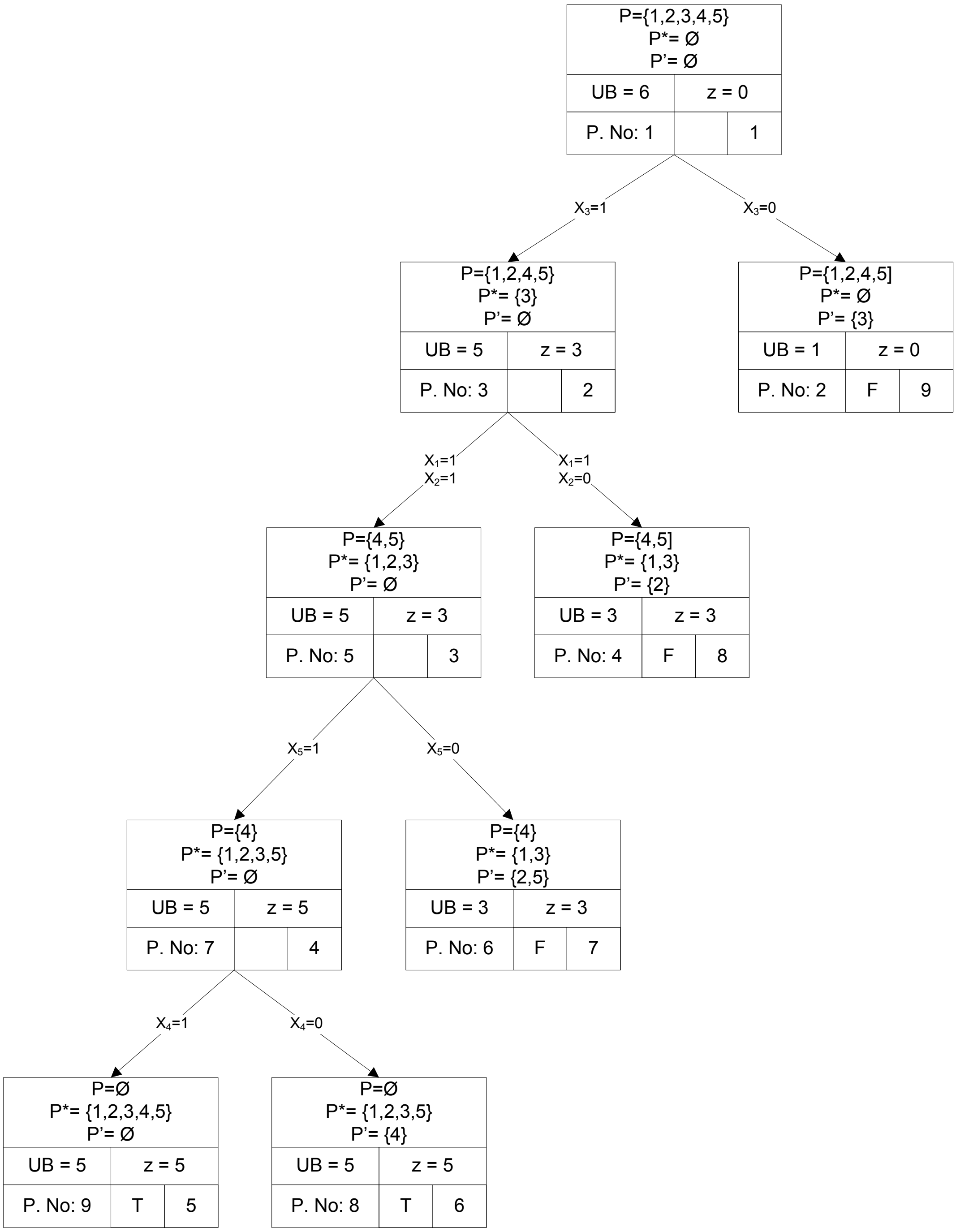
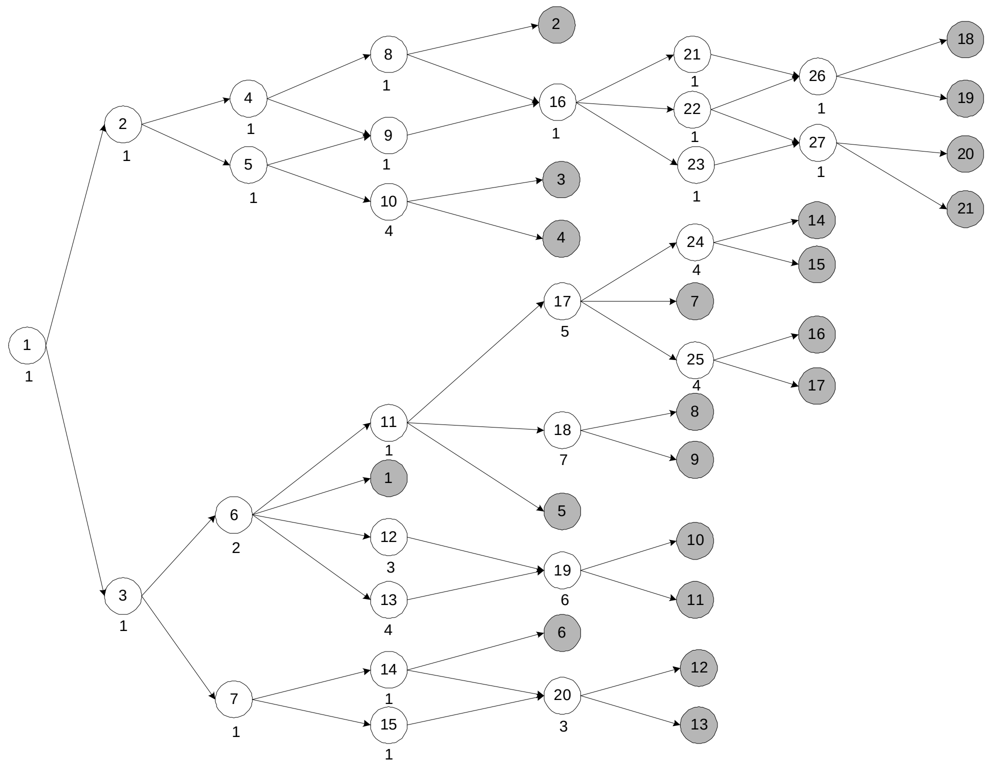


Figure A.4 Branch-and-Bound Search Tree for the example problem

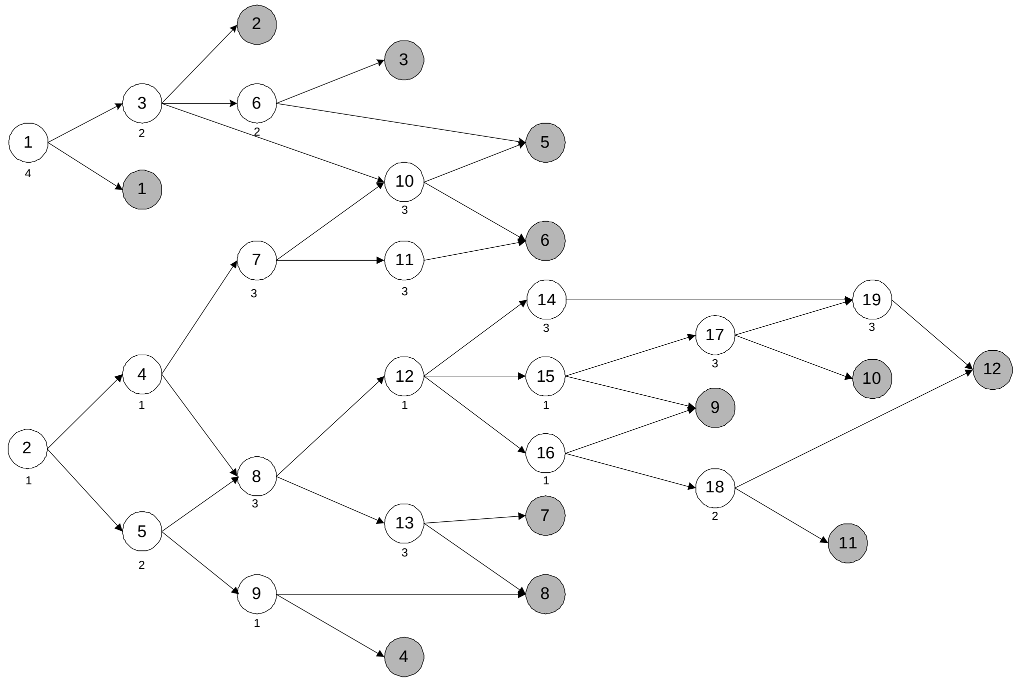
APPENDIX B

PRECEDENCE GRAPHS OF EXAMPLE PROBLEMS

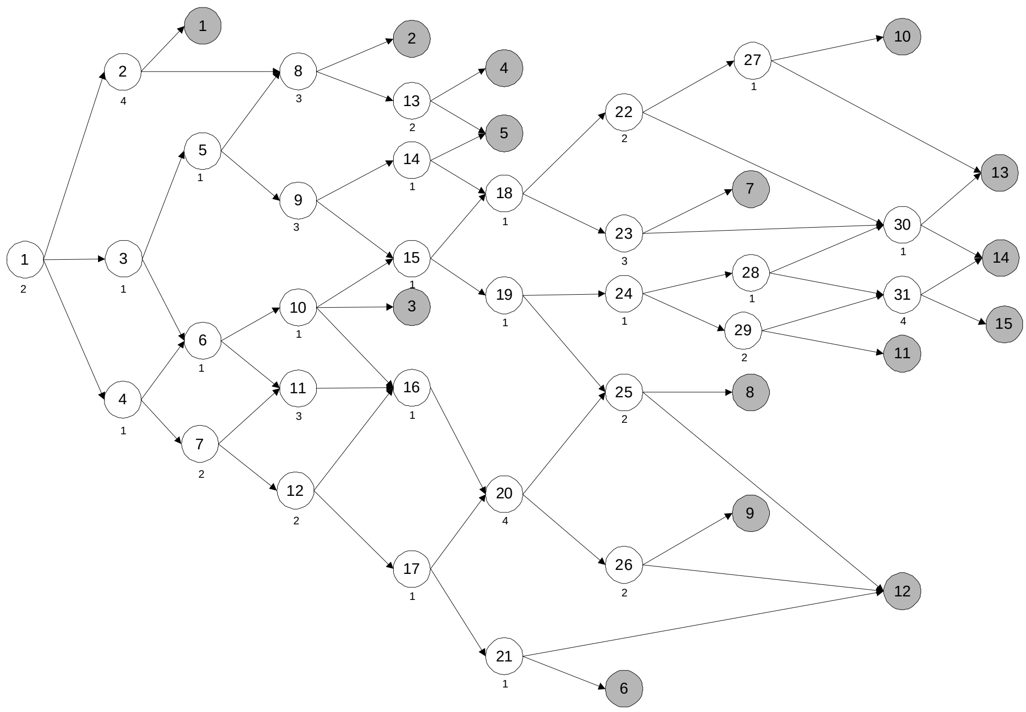
B.1 PRECEDENCE GRAPH OF YKA27T



B.2 PRECEDENCE GRAPH OF YKA19T



B.3 PRECEDENCE GRAPH OF YKA31T



APPENDIX C

EXPERIMENT RESULTS

Table C-1 Results for the initial Upper Bound experimentation

				NO UB	UB 1	UB 2	UB 3	UB BEST
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	CPU Time	CPU Time	CPU Time	CPU Time	CPU Time
BOWMAN8-1-0	8	4	1	0.0160	0.0000	0.0000	0.0150	0.0000
BOWMAN8-1-0	8	4	2	0.0000	0.0000	0.0000	0.0000	0.0000
BOWMAN8-1-0	8	4	3	0.0000	0.0000	0.0000	0.0000	0.0000
BOWMAN8-2-0	8	6	1	0.0160	0.0000	0.0000	0.0000	0.0000
BOWMAN8-2-0	8	6	2	0.0000	0.0000	0.0000	0.0160	0.0000
BOWMAN8-2-0	8	6	3	0.0000	0.0000	0.0000	0.0000	0.0000
BOWMAN8-3-0	8	8	1	0.0150	0.0000	0.0000	0.0000	0.0000
BOWMAN8-3-0	8	8	2	0.0160	0.0000	0.0000	0.0000	0.0000
BOWMAN8-3-0	8	8	3	0.0150	0.0000	0.0000	0.0000	0.0000
BUXEY-1-0	29	15	1	0.0160	0.0150	0.0000	0.0000	0.0000
BUXEY-1-0	29	15	2	0.1410	0.0160	0.0000	0.1560	0.0160
BUXEY-1-0	29	15	3	9.4380	0.1250	0.0780	11.7180	0.1410
BUXEY-2-0	29	22	1	0.0310	0.0000	0.0150	0.0160	0.0000
BUXEY-2-0	29	22	2	26.3280	0.6400	0.0630	16.8750	0.1100
BUXEY-2-0	29	22	3	1850.4280	3.9060	1.4680	2241.9620	2.5770
BUXEY-3-0	29	29	1	0.2660	0.0160	0.0150	0.2660	0.0160
BUXEY-3-0	29	29	2	1191.5030	1.7800	0.1700	622.5950	0.2800
BUXEY-3-0	29	29	3	5751.2730	368.2310	8.5850	6912.1640	17.5080
MERTENS-1-0	7	4	1	0.0160	0.0000	0.0000	0.0000	0.0000
MERTENS-1-0	7	4	2	0.0000	0.0000	0.0000	0.0000	0.0000
MERTENS-1-0	7	4	3	0.0000	0.0150	0.0000	0.0000	0.0000
MERTENS-2-0	7	6	1	0.0000	0.0000	0.0000	0.0000	0.0000
MERTENS-2-0	7	6	2	0.0000	0.0000	0.0000	0.0000	0.0000
MERTENS-2-0	7	6	3	0.0000	0.0000	0.0000	0.0150	0.0000

Table C-1 Results for the initial Upper Bound experimentation (Continued)

				NO UB	UB 1	UB 2	UB 3	UB BEST
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	CPU Time	CPU Time	CPU Time	CPU Time	CPU Time
MERTENS-3-0	7	7	1	0.0000	0.0000	0.0160	0.0000	0.0000
MERTENS-3-0	7	7	2	0.0000	0.0000	0.0000	0.0000	0.0000
MERTENS-3-0	7	7	3	0.0150	0.0000	0.0000	0.0000	0.0000
HESKIA-1-0	28	14	1	0.2470	0.0310	0.0000	0.1700	0.0150
HESKIA-1-0	28	14	2	1.8700	0.0160	0.0150	2.3340	0.0310
HESKIA-1-0	28	14	3	1.8690	0.0160	0.0150	2.3490	0.0310
HESKIA-2-0	28	21	1	46.5610	0.4330	0.0150	19.6260	0.0160
HESKIA-2-0	28	21	2	392.7720	0.1400	0.0470	372.2820	0.1250
HESKIA-2-0	28	21	3	394.4430	0.1410	0.0620	420.3200	0.1240
HESKIA-3-0	28	28	1	2184.7510	3.2400	0.0310	1492.1960	0.0630
HESKIA-3-0	28	28	2	4990.0480	2.9330	0.9210	5768.5850	1.8040
HESKIA-3-0	28	28	3	5010.5050	2.9450	0.9240	5880.3830	1.8000
JACKSON-1-0	11	6	1	0.0310	0.0000	0.0000	0.0000	0.0000
JACKSON-1-0	11	6	2	0.0000	0.0000	0.0000	0.0000	0.0000
JACKSON-1-0	11	6	3	0.0000	0.0000	0.0000	0.0000	0.0160
JACKSON-2-0	11	9	1	0.0000	0.0000	0.0000	0.0000	0.0150
JACKSON-2-0	11	9	2	0.0160	0.0310	0.0000	0.0160	0.0150
JACKSON-2-0	11	9	3	0.0160	0.0310	0.0000	0.0160	0.0000
JACKSON-3-0	11	11	1	0.0000	0.0150	0.0000	0.0000	0.0000
JACKSON-3-0	11	11	2	0.1100	0.0940	0.0000	0.0780	0.0150
JACKSON-3-0	11	11	3	0.1100	0.0780	0.0160	0.0780	0.0150
JAESCHKE-1-0	9	5	1	0.0160	0.0000	0.0000	0.0000	0.0000
JAESCHKE-1-0	9	5	2	0.0000	0.0000	0.0000	0.0000	0.0000
JAESCHKE-1-0	9	5	3	0.0000	0.0000	0.0000	0.0000	0.0000
JAESCHKE-2-0	9	7	1	0.0000	0.0000	0.0000	0.0000	0.0150
JAESCHKE-2-0	9	7	2	0.0000	0.0000	0.0000	0.0000	0.0000
JAESCHKE-2-0	9	7	3	0.0000	0.0000	0.0000	0.0160	0.0000
JAESCHKE-3-0	9	9	1	0.0000	0.0160	0.0000	0.0000	0.0000
JAESCHKE-3-0	9	9	2	0.0000	0.0000	0.0000	0.0000	0.0000
JAESCHKE-3-0	9	9	3	0.0000	0.0160	0.0000	0.0000	0.0000
LUTZ1-1-0	32	16	1	0.0310	0.0000	0.0000	0.0160	0.0000
LUTZ1-1-0	32	16	2	0.3130	0.0470	0.0150	0.3600	0.0160
LUTZ1-1-0	32	16	3	1.2520	0.0780	0.0320	1.0170	0.0620
LUTZ1-2-0	32	24	1	0.0630	0.0160	0.0000	0.0620	0.0160
LUTZ1-2-0	32	24	2	15.2300	2.5040	0.1720	10.0490	0.2350
LUTZ1-2-0	32	24	3	522.9460	1.9730	0.8140	346.0430	1.3930
LUTZ1-3-0	32	32	1	1.2520	0.0310	0.0160	0.5320	0.0470
LUTZ1-3-0	32	32	2	1411.2670	185.6530	1.9100	258.2800	3.4280
LUTZ1-3-0	32	32	3	6628.0710	57.2410	8.4370	7324.2570	13.9340
MANSOOR-1-0	11	6	1	0.0310	0.0000	0.0000	0.0000	0.0000
MANSOOR-1-0	11	6	2	0.0000	0.0000	0.0000	0.0000	0.0000
MANSOOR-1-0	11	6	3	0.0000	0.0000	0.0000	0.0000	0.0000
MANSOOR-2-0	11	9	1	0.0000	0.0000	0.0000	0.0000	0.0150

Table C-1 Results for the initial Upper Bound experimentation (Continued)

				NO UB	UB 1	UB 2	UB 3	UB BEST
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	CPU Time	CPU Time	CPU Time	CPU Time	CPU Time
MANSOOR-2-0	11	9	2	0.0160	0.0150	0.0000	0.0160	0.0000
MANSOOR-2-0	11	9	3	0.0460	0.0000	0.0160	0.0470	0.0150
MANSOOR-3-0	11	11	1	0.0000	0.0000	0.0000	0.0000	0.0000
MANSOOR-3-0	11	11	2	0.1090	0.0310	0.0000	0.0940	0.0000
MANSOOR-3-0	11	11	3	0.2340	0.0620	0.0000	0.1710	0.0150
MITCHELL-1-0	21	11	1	0.0160	0.0000	0.0000	0.0000	0.0150
MITCHELL-1-0	21	11	2	0.0160	0.0160	0.0000	0.0150	0.0000
MITCHELL-1-0	21	11	3	0.0470	0.0150	0.0000	0.0470	0.0150
MITCHELL-2-0	21	16	1	0.0160	0.0000	0.0000	0.0000	0.0150
MITCHELL-2-0	21	16	2	0.4680	0.3270	0.0160	0.3740	0.0150
MITCHELL-2-0	21	16	3	1.9480	0.5460	0.0160	2.3370	0.0460
MITCHELL-3-0	21	21	1	0.0780	0.0160	0.0000	0.0620	0.0000
MITCHELL-3-0	21	21	2	10.4110	7.5910	0.0620	10.3490	0.1250
MITCHELL-3-0	21	21	3	87.1880	2.0890	0.4360	91.1160	0.7950
ROSZIEG-1-0	25	13	1	0.0310	0.0000	0.0000	0.0150	0.0000
ROSZIEG-1-0	25	13	2	0.1100	0.0150	0.0000	0.1250	0.0000
ROSZIEG-1-0	25	13	3	0.2180	0.0160	0.0000	0.2650	0.0160
ROSZIEG-2-0	25	19	1	0.0470	0.0000	0.0000	0.0310	0.0000
ROSZIEG-2-0	25	19	2	2.5400	0.0470	0.0150	2.6970	0.0470
ROSZIEG-2-0	25	19	3	10.5670	0.1400	0.0630	9.4600	0.1400
ROSZIEG-3-0	25	25	1	0.8110	0.0310	0.0160	0.5300	0.0310
ROSZIEG-3-0	25	25	2	114.4170	2.4010	0.2180	121.7890	0.3890
ROSZIEG-3-0	25	25	3	955.5160	3.9120	0.9190	614.3690	1.7770
SAWYER30-1-0	30	15	1	0.0310	0.0150	0.0000	0.0160	0.0000
SAWYER30-1-0	30	15	2	1.1070	0.0310	0.0310	1.2310	0.0310
SAWYER30-1-0	30	15	3	9.5230	0.2960	0.0470	11.8140	0.0930
SAWYER30-2-0	30	23	1	0.4060	0.0460	0.0160	0.2650	0.0310
SAWYER30-2-0	30	23	2	115.0090	0.3900	0.0620	76.4810	0.0930
SAWYER30-2-0	30	23	3	3947.3730	13.4820	0.8410	4791.8360	1.7170
SAWYER30-3-0	30	30	1	9.2100	1.9980	0.0940	1.2800	0.0940
SAWYER30-3-0	30	30	2	5420.7530	709.0590	0.5170	6308.7920	0.9990
SAWYER30-3-0	30	30	3	6315.5230	800.3710	8.0900	7561.7100	16.6000
AKKSKT12-1-0	12	6	1	0.0000	0.0000	0.0000	0.0000	0.0000
AKKSKT12-1-0	12	6	2	0.0160	0.0000	0.0000	0.0000	0.0000
AKKSKT12-1-0	12	6	3	0.0160	0.0000	0.0000	0.0000	0.0150
AKKSKT12-2-0	12	9	1	0.0000	0.0000	0.0000	0.0000	0.0160
AKKSKT12-2-0	12	9	2	0.0160	0.0150	0.0000	0.0310	0.0000
AKKSKT12-2-0	12	9	3	0.0630	0.0000	0.0150	0.0320	0.0000
AKKSKT12-3-0	12	12	1	0.0310	0.0310	0.0000	0.0310	0.0000
AKKSKT12-3-0	12	12	2	0.2660	0.0470	0.0000	0.2340	0.0000
AKKSKT12-3-0	12	12	3	0.5300	0.0160	0.0000	0.3590	0.0000
AKO20T-1-0	15	8	1	0.0160	0.0000	0.0000	0.0000	0.0000
AKO20T-1-0	15	8	2	0.0000	0.0000	0.0160	0.0000	0.0000
AKO20T-1-0	15	8	3	0.0310	0.0000	0.0000	0.0310	0.0000
AKO20T-2-0	15	12	1	0.0000	0.0000	0.0150	0.0000	0.0000

Table C-1 Results for the initial Upper Bound experimentation (Continued)

				NO UB	UB 1	UB 2	UB 3	UB BEST
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	CPU Time	CPU Time	CPU Time	CPU Time	CPU Time
AKO20T-2-0	15	12	2	0.1410	0.0470	0.0000	0.1090	0.0160
AKO20T-2-0	15	12	3	0.6090	0.0310	0.0160	0.4060	0.0470
AKO20T-3-0	15	15	1	0.0000	0.0000	0.0000	0.0000	0.0000
AKO20T-3-0	15	15	2	0.7490	0.1560	0.0160	0.2650	0.0310
AKO20T-3-0	15	15	3	6.3250	0.2180	0.0630	3.2790	0.1090
AKO30T1-1-0	16	8	1	0.0000	0.0000	0.0000	0.0000	0.0160
AKO30T1-1-0	16	8	2	0.0000	0.0000	0.0150	0.0000	0.0000
AKO30T1-1-0	16	8	3	0.0000	0.0000	0.0000	0.0150	0.0000
AKO30T1-2-0	16	12	1	0.0160	0.0000	0.0000	0.0160	0.0150
AKO30T1-2-0	16	12	2	0.1410	0.1250	0.0000	0.1250	0.0150
AKO30T1-2-0	16	12	3	0.1410	0.1250	0.0000	0.1560	0.0000
AKO30T1-3-0	16	16	1	0.1100	0.0310	0.0160	0.0150	0.0000
AKO30T1-3-0	16	16	2	3.3410	1.9520	0.0160	1.0310	0.0310
AKO30T1-3-0	16	16	3	3.3420	1.9520	0.0150	1.4370	0.0160
AKO30T2-1-0	14	7	1	0.0160	0.0000	0.0000	0.0000	0.0000
AKO30T2-1-0	14	7	2	0.0000	0.0000	0.0000	0.0150	0.0000
AKO30T2-1-0	14	7	3	0.0000	0.0000	0.0160	0.0000	0.0000
AKO30T2-2-0	14	11	1	0.0000	0.0000	0.0000	0.0000	0.0000
AKO30T2-2-0	14	11	2	0.0620	0.0160	0.0000	0.0620	0.0000
AKO30T2-2-0	14	11	3	0.0620	0.0160	0.0150	0.0630	0.0150
AKO30T2-3-0	14	14	1	0.0000	0.0160	0.0000	0.0000	0.0000
AKO30T2-3-0	14	14	2	0.6720	0.0780	0.0000	0.2030	0.0150
AKO30T2-3-0	14	14	3	0.6720	0.0780	0.0000	0.4220	0.0150
GUN17T-1-0	17	9	1	0.0000	0.0000	0.0000	0.0000	0.0000
GUN17T-1-0	17	9	2	0.0160	0.0160	0.0000	0.0150	0.0000
GUN17T-1-0	17	9	3	0.0310	0.0160	0.0150	0.0320	0.0160
GUN17T-2-0	17	13	1	0.0160	0.0150	0.0160	0.0150	0.0160
GUN17T-2-0	17	13	2	0.3280	0.0470	0.0000	0.2650	0.0150
GUN17T-2-0	17	13	3	0.7030	0.0630	0.0150	0.8120	0.0160
GUN17T-3-0	17	17	1	0.4370	0.1560	0.0000	0.2190	0.0000
GUN17T-3-0	17	17	2	7.4650	0.4680	0.0160	4.2470	0.0310
GUN17T-3-0	17	17	3	15.2100	1.4990	0.0310	15.6470	0.0310
GUN8T-1-0	8	4	1	0.0000	0.0000	0.0000	0.0150	0.0000
GUN8T-1-0	8	4	2	0.0000	0.0000	0.0000	0.0000	0.0000
GUN8T-1-0	8	4	3	0.0000	0.0000	0.0000	0.0000	0.0000
GUN8T-2-0	8	6	1	0.0000	0.0000	0.0160	0.0000	0.0000
GUN8T-2-0	8	6	2	0.0000	0.0000	0.0000	0.0000	0.0000
GUN8T-2-0	8	6	3	0.0000	0.0000	0.0000	0.0000	0.0160
GUN8T-3-0	8	8	1	0.0000	0.0000	0.0000	0.0000	0.0000
GUN8T-3-0	8	8	2	0.0000	0.0000	0.0000	0.0000	0.0000
GUN8T-3-0	8	8	3	0.0150	0.0000	0.0000	0.0000	0.0000
LAM20T-1-0	9	5	1	0.0000	0.0000	0.0000	0.0000	0.0000
LAM20T-1-0	9	5	2	0.0000	0.0000	0.0000	0.0150	0.0000
LAM20T-1-0	9	5	3	0.0000	0.0000	0.0000	0.0000	0.0160
LAM20T-2-0	9	7	1	0.0000	0.0000	0.0000	0.0000	0.0000

Table C-1 Results for the initial Upper Bound experimentation (Continued)

				NO UB	UB 1	UB 2	UB 3	UB BEST
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	CPU Time	CPU Time	CPU Time	CPU Time	CPU Time
LAM20T-2-0	9	7	2	0.0000	0.0160	0.0000	0.0000	0.0000
LAM20T-2-0	9	7	3	0.0150	0.0000	0.0000	0.0160	0.0000
LAM20T-3-0	9	9	1	0.0000	0.0000	0.0000	0.0000	0.0000
LAM20T-3-0	9	9	2	0.0470	0.0150	0.0000	0.0160	0.0000
LAM20T-3-0	9	9	3	0.0470	0.0000	0.0150	0.0160	0.0000
LAM30T-1-0	9	5	1	0.0000	0.0000	0.0000	0.0000	0.0000
LAM30T-1-0	9	5	2	0.0000	0.0160	0.0000	0.0000	0.0000
LAM30T-1-0	9	5	3	0.0000	0.0000	0.0000	0.0000	0.0000
LAM30T-2-0	9	7	1	0.0000	0.0000	0.0000	0.0150	0.0000
LAM30T-2-0	9	7	2	0.0000	0.0000	0.0000	0.0000	0.0000
LAM30T-2-0	9	7	3	0.0000	0.0160	0.0000	0.0000	0.0000
LAM30T-3-0	9	9	1	0.0000	0.0000	0.0000	0.0160	0.0000
LAM30T-3-0	9	9	2	0.0000	0.0000	0.0000	0.0150	0.0000
LAM30T-3-0	9	9	3	0.0160	0.0150	0.0000	0.0000	0.0000
MAS30T-1-0	30	15	1	0.1250	0.0160	0.0150	0.1410	0.0000
MAS30T-1-0	30	15	2	2.2170	0.0160	0.0160	2.7010	0.0160
MAS30T-1-0	30	15	3	9.3540	0.0150	0.0310	11.6650	0.0310
MAS30T-2-0	30	23	1	13.3360	0.0930	0.0320	9.0880	0.0470
MAS30T-2-0	30	23	2	467.7650	0.1880	0.2810	327.9740	0.3430
MAS30T-2-0	30	23	3	3942.3890	0.6410	0.6410	4736.3860	0.9250
MAS30T-3-0	30	30	1	302.1560	0.4380	0.1730	131.6800	0.2510
MAS30T-3-0	30	30	2	5574.3470	3.3280	3.0160	6333.3480	4.2660
MAS30T-3-0	30	30	3	6288.0160	16.0310	11.5470	7338.4790	20.7970
MGG7T-1-0	7	4	1	0.0160	0.0000	0.0150	0.0000	0.0000
MGG7T-1-0	7	4	2	0.0000	0.0000	0.0000	0.0000	0.0000
MGG7T-1-0	7	4	3	0.0000	0.0000	0.0000	0.0000	0.0000
MGG7T-2-0	7	6	1	0.0160	0.0000	0.0000	0.0000	0.0000
MGG7T-2-0	7	6	2	0.0000	0.0000	0.0160	0.0000	0.0000
MGG7T-2-0	7	6	3	0.0000	0.0000	0.0000	0.0000	0.0150
MGG7T-3-0	7	7	1	0.0000	0.0000	0.0000	0.0000	0.0000
MGG7T-3-0	7	7	2	0.0160	0.0000	0.0000	0.0000	0.0000
MGG7T-3-0	7	7	3	0.0150	0.0000	0.0000	0.0000	0.0000
WANG18T-1-0	20	10	1	0.0320	0.0000	0.0150	0.0000	0.0000
WANG18T-1-0	20	10	2	0.1570	0.0000	0.0150	0.1560	0.0000
WANG18T-1-0	20	10	3	0.1410	0.0150	0.0160	0.1870	0.0160
WANG18T-2-0	20	15	1	0.0630	0.0150	0.0000	0.0630	0.0150
WANG18T-2-0	20	15	2	7.2030	0.0320	0.0620	8.7500	0.0620
WANG18T-2-0	20	15	3	7.2030	0.0470	0.0470	8.7660	0.0620
WANG18T-3-0	20	20	1	0.5630	0.3750	0.0150	0.6570	0.0310
WANG18T-3-0	20	20	2	324.2370	1.4680	0.5470	129.0950	0.6880
WANG18T-3-0	20	20	3	324.1740	1.4680	0.5470	136.8290	0.7190
YKA19T-1-0	19	10	1	0.0470	0.0160	0.0000	0.0150	0.0000
YKA19T-1-0	19	10	2	0.1400	0.0000	0.0160	0.1560	0.0000
YKA19T-1-0	19	10	3	0.1410	0.0000	0.0150	0.1560	0.0000
YKA19T-2-0	19	15	1	0.8440	0.0470	0.0150	0.3280	0.0150

Table C-1 Results for the initial Upper Bound experimentation (Continued)

				NO UB	UB 1	UB 2	UB 3	UB BEST
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	CPU Time	CPU Time	CPU Time	CPU Time	CPU Time
YKA19T-2-0	19	15	2	7.0780	0.0940	0.0160	2.1090	0.0470
YKA19T-2-0	19	15	3	7.0780	0.0780	0.0160	2.2500	0.0310
YKA19T-3-0	19	19	1	17.8440	3.0320	0.0310	1.4220	0.0470
YKA19T-3-0	19	19	2	148.5010	4.8750	0.1250	72.9230	0.1870
YKA19T-3-0	19	19	3	148.5010	4.8910	0.1250	73.3750	0.1870
YKA27T-1-0	27	14	1	2.0310	0.0790	0.1090	0.6870	0.0310
YKA27T-1-0	27	14	2	4.0620	0.1720	0.1250	2.1570	0.0780
YKA27T-1-0	27	14	3	4.0620	0.1720	0.1250	2.1560	0.0790
YKA27T-2-0	27	21	1	405.6900	0.2030	0.0780	55.5630	0.1090
YKA27T-2-0	27	21	2	823.7550	0.6720	0.1410	153.1260	0.2340
YKA27T-2-0	27	21	3	820.8650	0.6720	0.1400	155.6730	0.2500
YKA27T-3-0	27	27	1	4620.5920	0.2650	0.7660	220.6110	0.3910
YKA27T-3-0	27	27	2	4690.2910	0.2660	0.7810	576.6290	0.4530
YKA27T-3-0	27	27	3	4688.0920	0.2500	0.7810	587.7850	0.4370
YKA31T-1-0	31	16	1	0.3280	0.0470	0.0160	0.1560	0.0150
YKA31T-1-0	31	16	2	20.9220	0.1100	0.0780	13.7340	0.1250
YKA31T-1-0	31	16	3	20.9220	0.1250	0.0780	14.9370	0.1250
YKA31T-2-0	31	24	1	29.9380	0.5780	0.0310	15.5310	0.0470
YKA31T-2-0	31	24	2	4271.6520	2.7970	0.3750	3633.0710	0.5780
YKA31T-2-0	31	24	3	4283.3870	2.8440	0.3590	3991.5570	0.5780
YKA31T-3-0	31	31	1	2673.5640	4.4070	0.1250	293.3760	0.2340
YKA31T-3-0	31	31	2	7009.3860	18.4460	2.1710	8013.5910	4.3270
YKA31T-3-0	31	31	3	7004.9440	18.4460	2.1710	8195.8690	4.3270

Table C-2 Results for the initial Upper Bound experimentation

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
BOWMAN8-1-0	8	4	1	1	1	1	1	1	1	1	1	1	1
BOWMAN8-1-0	8	4	2	3	1	3	1	3	1	3	1	3	1
BOWMAN8-1-0	8	4	3	7	1	5	1	5	1	7	1	5	1
BOWMAN8-2-0	8	6	1	7	3	7	3	5	3	7	3	5	3
BOWMAN8-2-0	8	6	2	15	3	7	3	7	3	15	3	7	3
BOWMAN8-2-0	8	6	3	31	3	11	3	9	3	25	3	9	3
BOWMAN8-3-0	8	8	1	31	5	17	5	11	5	21	5	11	5
BOWMAN8-3-0	8	8	2	63	5	33	5	15	5	45	5	15	5
BOWMAN8-3-0	8	8	3	127	13	45	13	21	13	101	13	21	13
BUXEY-1-0	29	15	1	15	1	9	1	7	1	15	1	7	1
BUXEY-1-0	29	15	2	1023	19	61	19	31	19	995	19	27	19
BUXEY-1-0	29	15	3	65535	21	529	21	329	21	65535	21	285	21
BUXEY-2-0	29	22	1	127	7	25	7	15	7	127	7	15	7
BUXEY-2-0	29	22	2	131071	31	2437	31	203	31	70921	31	177	31
BUXEY-2-0	29	22	3	8388607	1155	11609	273	3715	81	8388441	1155	3583	81
BUXEY-3-0	29	29	1	1023	9	39	9	31	9	983	9	27	9
BUXEY-3-0	29	29	2	4194303	31	4567	31	397	31	1886171	31	343	31
BUXEY-3-0	29	29	3	16777216	6181	867799	389	18025	113	16777216	5797	17935	113
MERTENS-1-0	7	4	1	7	5	7	5	5	5	7	5	5	5
MERTENS-1-0	7	4	2	7	5	7	5	5	5	7	5	5	5
MERTENS-1-0	7	4	3	15	5	9	5	7	5	13	5	7	5
MERTENS-2-0	7	6	1	15	3	11	3	9	3	9	3	9	3
MERTENS-2-0	7	6	2	31	3	17	3	17	3	17	3	17	3
MERTENS-2-0	7	6	3	63	3	29	3	25	3	35	3	25	3
MERTENS-3-0	7	7	1	31	1	17	1	9	1	17	1	9	1
MERTENS-3-0	7	7	2	63	1	21	1	11	1	53	1	11	1

Table C-2 Results for the initial Upper Bound experimentation (Continued)

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
HESKIA-1-0	28	14	2	16383	21	59	21	91	21	16383	21	53	21
HESKIA-1-0	28	14	3	16383	21	59	21	91	21	16383	21	53	21
HESKIA-2-0	28	21	1	262143	23	1617	23	43	23	93549	23	43	23
HESKIA-2-0	28	21	2	2097151	23	433	23	185	23	1636749	23	177	23
HESKIA-2-0	28	21	3	2097151	23	433	23	185	23	1846041	23	177	23
HESKIA-3-0	28	28	1	8388607	415	8305	41	73	35	4931001	227	69	35
HESKIA-3-0	28	28	2	16777216	12319	6561	39	2081	53	16777216	6513	1931	39
HESKIA-3-0	28	28	3	16777216	12319	6561	39	2081	53	16777216	6659	1931	39
JACKSON-1-0	11	6	1	3	1	3	1	3	1	3	1	3	1
JACKSON-1-0	11	6	2	63	7	47	7	15	7	63	7	13	7
JACKSON-1-0	11	6	3	63	7	47	7	15	7	63	7	13	7
JACKSON-2-0	11	9	1	15	7	15	7	7	7	15	7	7	7
JACKSON-2-0	11	9	2	511	11	439	11	23	11	367	11	19	11
JACKSON-2-0	11	9	3	511	11	439	11	23	11	367	11	19	11
JACKSON-3-0	11	11	1	31	3	31	3	9	3	21	3	9	3
JACKSON-3-0	11	11	2	2047	17	1157	17	43	17	1183	17	37	17
JACKSON-3-0	11	11	3	2047	17	1157	17	43	17	1205	17	37	17
JAESCHKE-1-0	9	5	1	1	1	1	1	1	1	1	1	1	1
JAESCHKE-1-0	9	5	2	3	1	3	1	3	1	3	1	3	1
JAESCHKE-1-0	9	5	3	15	1	13	1	7	1	15	1	7	1
JAESCHKE-2-0	9	7	1	3	3	3	3	3	3	3	3	3	3
JAESCHKE-2-0	9	7	2	15	3	9	3	9	3	11	3	9	3
JAESCHKE-2-0	9	7	3	63	3	49	3	17	3	63	3	17	3
JAESCHKE-3-0	9	9	1	15	5	15	5	7	5	11	5	7	5
JAESCHKE-3-0	9	9	2	63	5	37	5	19	5	27	5	19	5
JAESCHKE-3-0	9	9	3	255	5	75	5	31	5	165	5	31	5
LUTZ1-1-0	32	16	1	31	5	11	5	9	5	31	5	9	5
LUTZ1-1-0	32	16	2	2047	19	283	19	75	19	2033	19	53	19

Table C-2 Results for the initial Upper Bound experimentation (Continued)

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
LUTZ1-1-0	32	16	3	8191	19	333	19	165	19	5503	19	139	19
LUTZ1-2-0	32	24	1	255	11	17	11	17	11	229	11	17	11
LUTZ1-2-0	32	24	2	65535	11	8615	11	499	11	36767	11	349	11
LUTZ1-2-0	32	24	3	2097151	11	5299	11	2115	11	1171223	11	1831	11
LUTZ1-3-0	32	32	1	4095	13	55	13	65	13	1505	13	51	13
LUTZ1-3-0	32	32	2	4194303	31	454297	31	3929	31	646975	31	3767	31
LUTZ1-3-0	32	32	3	16777216	31	110435	31	15845	31	16777216	31	13053	31
MANSOOR-1-0	11	6	1	1	1	1	1	1	1	1	1	1	1
MANSOOR-1-0	11	6	2	63	9	17	9	11	9	37	9	11	9
MANSOOR-1-0	11	6	3	127	9	27	9	21	9	85	9	17	9
MANSOOR-2-0	11	9	1	7	1	7	1	5	1	7	1	5	1
MANSOOR-2-0	11	9	2	511	15	121	15	19	15	367	15	19	15
MANSOOR-2-0	11	9	3	1023	15	131	15	35	15	789	15	27	15
MANSOOR-3-0	11	11	1	15	1	15	1	11	1	15	1	11	1
MANSOOR-3-0	11	11	2	2047	19	375	19	23	19	1251	19	21	19
MANSOOR-3-0	11	11	3	4095	19	641	19	43	19	2561	19	43	19
MITCHELL-1-0	21	11	1	15	5	7	5	7	5	15	5	7	5
MITCHELL-1-0	21	11	2	255	13	93	13	25	13	255	13	21	13
MITCHELL-1-0	21	11	3	511	13	91	13	49	13	511	13	45	13
MITCHELL-2-0	21	16	1	31	7	13	7	9	7	23	7	9	7
MITCHELL-2-0	21	16	2	4095	21	2417	21	67	21	2777	21	57	21
MITCHELL-2-0	21	16	3	16383	25	3505	25	121	25	16383	25	117	25
MITCHELL-3-0	21	21	1	511	7	45	7	49	7	309	7	45	7
MITCHELL-3-0	21	21	2	65535	23	40531	23	323	23	55653	23	249	23
MITCHELL-3-0	21	21	3	524287	25	9021	25	1777	25	463767	25	1679	25
ROSZIEG-1-0	25	13	1	31	7	15	7	11	7	31	7	11	7
ROSZIEG-1-0	25	13	2	1023	15	129	15	31	15	1019	15	31	15
ROSZIEG-1-0	25	13	3	2047	15	55	15	47	15	2047	15	45	15

Table C-2 Results for the initial Upper Bound experimentation (Continued)

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
ROSZIEG-2-0	25	19	1	255	7	31	7	15	7	185	7	15	7
ROSZIEG-2-0	25	19	2	16383	19	173	19	113	19	14579	19	101	19
ROSZIEG-2-0	25	19	3	65535	19	561	19	283	19	49151	19	271	19
ROSZIEG-3-0	25	25	1	4095	9	113	9	49	9	2299	9	49	9
ROSZIEG-3-0	25	25	2	524287	27	8475	27	659	27	478321	27	645	27
ROSZIEG-3-0	25	25	3	4194303	27	12055	27	2733	27	2296123	27	2683	27
SAWYER30-1-0	30	15	1	63	9	31	9	11	9	63	9	11	9
SAWYER30-1-0	30	15	2	8191	21	185	21	93	21	7529	21	49	21
SAWYER30-1-0	30	15	3	65535	533	1415	117	183	45	65535	533	169	43
SAWYER30-2-0	30	23	1	2047	17	153	17	75	17	1137	17	63	17
SAWYER30-2-0	30	23	2	524287	25	1223	25	195	25	289971	25	145	25
SAWYER30-2-0	30	23	3	16777215	29	40211	29	2245	29	16749877	29	2181	29
SAWYER30-3-0	30	30	1	32767	8211	5987	2133	235	75	3839	1519	139	61
SAWYER30-3-0	30	30	2	16777216	32815	1790815	14123	1087	121	16777216	19245	1017	115
SAWYER30-3-0	30	30	3	16777216	1148613	1767417	47527	15835	749	16777216	948213	15753	749
AKKSKT12-1-0	12	6	1	31	9	19	9	9	9	25	9	9	9
AKKSKT12-1-0	12	6	2	63	9	23	9	11	9	63	9	11	9
AKKSKT12-1-0	12	6	3	127	9	15	9	15	9	97	9	13	9
AKKSKT12-2-0	12	9	1	63	11	39	11	13	11	47	11	13	11
AKKSKT12-2-0	12	9	2	511	11	239	11	23	11	383	11	17	11
AKKSKT12-2-0	12	9	3	1023	11	161	11	37	11	543	11	19	11
AKKSKT12-3-0	12	12	1	511	11	271	11	25	11	337	11	17	11
AKKSKT12-3-0	12	12	2	4095	11	483	11	51	11	2835	11	33	11
AKKSKT12-3-0	12	12	3	8191	267	97	37	81	15	4485	135	37	13
AKO20T-1-0	15	8	1	3	3	3	3	3	3	3	3	3	3
AKO20T-1-0	15	8	2	127	3	17	3	15	3	103	3	13	3
AKO20T-1-0	15	8	3	511	3	19	3	23	3	511	3	19	3
AKO20T-2-0	15	12	1	15	3	9	3	7	3	9	3	7	3

Table C-2 Results for the initial Upper Bound experimentation (Continued)

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
AKO20T-2-0	15	12	2	2047	1539	435	377	83	65	1227	919	75	57
AKO20T-2-0	15	12	3	8191	4199	305	229	183	129	4331	2181	145	97
AKO20T-3-0	15	15	1	63	1	49	1	13	1	39	1	13	1
AKO20T-3-0	15	15	2	8191	1	1261	1	145	1	2451	1	113	1
AKO20T-3-0	15	15	3	65535	1117	1571	225	507	71	27803	465	383	67
AKO30T1-1-0	16	8	1	7	5	7	5	5	5	5	5	5	5
AKO30T1-1-0	16	8	2	127	5	85	5	23	5	111	5	21	5
AKO30T1-1-0	16	8	3	127	5	85	5	23	5	111	5	21	5
AKO30T1-2-0	16	12	1	127	7	45	7	17	7	117	7	17	7
AKO30T1-2-0	16	12	2	2047	11	1339	11	69	11	1441	11	57	11
AKO30T1-2-0	16	12	3	2047	11	1339	11	69	11	1729	11	57	11
AKO30T1-3-0	16	16	1	1023	75	293	45	67	31	165	33	57	31
AKO30T1-3-0	16	16	2	32767	39	15601	39	95	31	8025	31	95	31
AKO30T1-3-0	16	16	3	32767	39	15601	39	95	31	11565	31	95	31
AKO30T2-1-0	14	7	1	7	1	5	1	5	1	7	1	5	1
AKO30T2-1-0	14	7	2	63	1	49	1	19	1	63	1	17	1
AKO30T2-1-0	14	7	3	63	1	49	1	19	1	63	1	17	1
AKO30T2-2-0	14	11	1	15	3	9	3	7	3	11	3	7	3
AKO30T2-2-0	14	11	2	1023	13	243	13	29	13	769	13	23	13
AKO30T2-2-0	14	11	3	1023	13	243	13	29	13	839	13	25	13
AKO30T2-3-0	14	14	1	127	3	29	3	17	3	49	3	13	3
AKO30T2-3-0	14	14	2	8191	651	707	175	59	31	2065	389	51	31
AKO30T2-3-0	14	14	3	8191	651	707	175	59	31	4201	471	51	31
GUN17T-1-0	17	9	1	15	7	15	7	7	7	15	7	7	7
GUN17T-1-0	17	9	2	255	7	99	7	55	7	255	7	27	7
GUN17T-1-0	17	9	3	511	17	215	17	73	17	511	17	39	17
GUN17T-2-0	17	13	1	255	15	255	15	15	15	191	15	15	15
GUN17T-2-0	17	13	2	4095	15	349	15	43	15	2611	15	25	15

Table C-2 Results for the initial Upper Bound experimentation (Continued)

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
AKO20T-2-0	15	12	2	2047	1539	435	377	83	65	1227	919	75	57
AKO20T-2-0	15	12	3	8191	4199	305	229	183	129	4331	2181	145	97
AKO20T-3-0	15	15	1	63	1	49	1	13	1	39	1	13	1
AKO20T-3-0	15	15	2	8191	1	1261	1	145	1	2451	1	113	1
AKO20T-3-0	15	15	3	65535	1117	1571	225	507	71	27803	465	383	67
AKO30T1-1-0	16	8	1	7	5	7	5	5	5	5	5	5	5
AKO30T1-1-0	16	8	2	127	5	85	5	23	5	111	5	21	5
AKO30T1-1-0	16	8	3	127	5	85	5	23	5	111	5	21	5
AKO30T1-2-0	16	12	1	127	7	45	7	17	7	117	7	17	7
AKO30T1-2-0	16	12	2	2047	11	1339	11	69	11	1441	11	57	11
AKO30T1-2-0	16	12	3	2047	11	1339	11	69	11	1729	11	57	11
AKO30T1-3-0	16	16	1	1023	75	293	45	67	31	165	33	57	31
AKO30T1-3-0	16	16	2	32767	39	15601	39	95	31	8025	31	95	31
AKO30T1-3-0	16	16	3	32767	39	15601	39	95	31	11565	31	95	31
AKO30T2-1-0	14	7	1	7	1	5	1	5	1	7	1	5	1
AKO30T2-1-0	14	7	2	63	1	49	1	19	1	63	1	17	1
AKO30T2-1-0	14	7	3	63	1	49	1	19	1	63	1	17	1
AKO30T2-2-0	14	11	1	15	3	9	3	7	3	11	3	7	3
AKO30T2-2-0	14	11	2	1023	13	243	13	29	13	769	13	23	13
AKO30T2-2-0	14	11	3	1023	13	243	13	29	13	839	13	25	13
AKO30T2-3-0	14	14	1	127	3	29	3	17	3	49	3	13	3
AKO30T2-3-0	14	14	2	8191	651	707	175	59	31	2065	389	51	31
AKO30T2-3-0	14	14	3	8191	651	707	175	59	31	4201	471	51	31
GUN17T-1-0	17	9	1	15	7	15	7	7	7	15	7	7	7
GUN17T-1-0	17	9	2	255	7	99	7	55	7	255	7	27	7
GUN17T-1-0	17	9	3	511	17	215	17	73	17	511	17	39	17
GUN17T-2-0	17	13	1	255	15	255	15	15	15	191	15	15	15
GUN17T-2-0	17	13	2	4095	15	349	15	43	15	2611	15	25	15

Table C-2 Results for the initial Upper Bound experimentation (Continued)

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
GUN17T-2-0	17	13	3	8191	15	613	15	85	15	7791	15	39	15
GUN17T-3-0	17	17	1	4095	19	1211	19	27	19	1721	19	27	19
GUN17T-3-0	17	17	2	65535	19	3073	19	75	19	31461	19	71	19
GUN17T-3-0	17	17	3	131071	531	10189	271	115	27	112761	347	111	27
GUN8T-1-0	8	4	1	1	1	1	1	1	1	1	1	1	1
GUN8T-1-0	8	4	2	3	3	3	3	3	3	3	3	3	3
GUN8T-1-0	8	4	3	15	3	7	3	9	3	15	3	7	3
GUN8T-2-0	8	6	1	3	3	3	3	3	3	3	3	3	3
GUN8T-2-0	8	6	2	7	5	7	5	5	5	7	5	5	5
GUN8T-2-0	8	6	3	31	5	9	5	9	5	27	5	9	5
GUN8T-3-0	8	8	1	7	3	5	3	5	3	7	3	5	3
GUN8T-3-0	8	8	2	31	5	13	5	11	5	15	5	9	5
GUN8T-3-0	8	8	3	127	11	45	11	29	11	97	11	27	11
LAM20T-1-0	9	5	1	1	1	1	1	1	1	1	1	1	1
LAM20T-1-0	9	5	2	63	9	35	9	21	9	63	9	11	9
LAM20T-1-0	9	5	3	63	9	35	9	21	9	63	9	11	9
LAM20T-2-0	9	7	1	3	3	3	3	3	3	3	3	3	3
LAM20T-2-0	9	7	2	255	9	27	9	21	9	139	9	15	9
LAM20T-2-0	9	7	3	255	9	27	9	21	9	167	9	15	9
LAM20T-3-0	9	9	1	15	3	7	3	7	3	15	3	7	3
LAM20T-3-0	9	9	2	1023	3	73	3	45	3	259	3	31	3
LAM20T-3-0	9	9	3	1023	3	73	3	45	3	279	3	31	3
LAM30T-1-0	9	5	1	1	1	1	1	1	1	1	1	1	1
LAM30T-1-0	9	5	2	7	1	5	1	5	1	7	1	5	1
LAM30T-1-0	9	5	3	31	1	17	1	15	1	27	1	15	1
LAM30T-2-0	9	7	1	3	3	3	3	3	3	3	3	3	3
LAM30T-2-0	9	7	2	31	3	11	3	9	3	21	3	9	3
LAM30T-2-0	9	7	3	127	3	21	3	19	3	69	3	13	3

Table C-2 Results for the initial Upper Bound experimentation (Continued)

				NO UB		UB 1		UB 2		UB 3		UB BEST	
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node	# of Nodes	Optimal Node
LAM30T-3-0	9	9	1	15	7	15	7	7	7	13	7	7	7
LAM30T-3-0	9	9	2	127	7	27	7	21	7	95	7	21	7
LAM30T-3-0	9	9	3	511	13	33	13	23	13	197	13	23	13
MAS30T-1-0	30	15	1	1023	15	117	15	19	15	1023	15	19	15
MAS30T-1-0	30	15	2	16383	15	79	15	75	15	16383	15	29	15
MAS30T-1-0	30	15	3	65535	15	45	15	155	15	65535	15	45	15
MAS30T-2-0	30	23	1	65535	85	323	25	107	25	37763	85	83	25
MAS30T-2-0	30	23	2	2097151	12821	553	127	825	179	1230407	11831	455	89
MAS30T-2-0	30	23	3	16777215	114469	1679	259	1699	241	16773501	114469	1137	159
MAS30T-3-0	30	30	1	1048575	283	1081	43	385	43	398255	73	305	43
MAS30T-3-0	30	30	2	16777216	262171	6925	497	6233	291	16777216	96311	4367	215
MAS30T-3-0	30	30	3	16777216	2197179	31039	625	22797	351	16777216	1818737	19431	259
MGG7T-1-0	7	4	1	7	5	7	5	5	5	7	5	5	5
MGG7T-1-0	7	4	2	15	7	9	7	7	7	15	7	7	7
MGG7T-1-0	7	4	3	15	7	9	7	7	7	15	7	7	7
MGG7T-2-0	7	6	1	15	7	9	7	7	7	9	7	7	7
MGG7T-2-0	7	6	2	63	9	51	9	13	9	41	9	11	9
MGG7T-2-0	7	6	3	63	9	51	9	13	9	43	9	11	9
MGG7T-3-0	7	7	1	31	7	11	7	9	7	11	7	9	7
MGG7T-3-0	7	7	2	127	11	77	11	13	11	57	11	13	11
MGG7T-3-0	7	7	3	127	11	77	11	13	11	61	11	13	11
WANG18T-1-0	20	10	1	63	9	37	9	11	9	63	9	11	9
WANG18T-1-0	20	10	2	2047	9	107	9	83	9	1717	9	31	9
WANG18T-1-0	20	10	3	2047	9	107	9	83	9	2035	9	31	9
WANG18T-2-0	20	15	1	511	13	161	13	19	13	511	13	19	13
WANG18T-2-0	20	15	2	65535	13	251	13	291	13	65497	13	157	13
WANG18T-2-0	20	15	3	65535	13	251	13	291	13	65535	13	157	13
WANG18T-3-0	20	20	1	4095	19	2277	19	91	19	4061	19	79	19

Table C-2 Results for the initial Upper Bound experimentation (Continued)

WANG18T-3-0	20	20	2	2097151	29	6985	29	2329	29	678285	29	1429	29
WANG18T-3-0	20	20	3	2097151	29	6985	29	2329	29	721483	29	1461	29
YKA19T-1-0	19	10	1	511	13	37	13	21	13	239	13	17	13
YKA19T-1-0	19	10	2	2047	13	47	13	27	13	1795	13	21	13
YKA19T-1-0	19	10	3	2047	13	47	13	27	13	1809	13	21	13
YKA19T-2-0	19	15	1	8191	13	325	13	61	13	2687	13	59	13
YKA19T-2-0	19	15	2	65535	13	567	13	159	13	15231	13	129	13
YKA19T-2-0	19	15	3	65535	13	567	13	159	13	16273	13	129	13
YKA19T-3-0	19	19	1	131071	23	17109	23	159	23	8463	23	87	23
YKA19T-3-0	19	19	2	1048575	25	26045	25	595	25	424831	25	425	25
YKA19T-3-0	19	19	3	1048575	25	26045	25	595	25	427491	25	425	25
YKA27T-1-0	27	14	1	16383	11	435	11	545	11	4551	11	79	11
YKA27T-1-0	27	14	2	32767	11	879	11	677	11	13823	11	173	11
YKA27T-1-0	27	14	3	32767	11	879	11	677	11	13823	11	205	11
YKA27T-2-0	27	21	1	2097151	23	639	23	287	23	233661	23	175	23
YKA27T-2-0	27	21	2	4194303	23	2091	23	457	23	639177	23	357	23
YKA27T-2-0	27	21	3	4194303	23	2091	23	457	23	650411	23	357	23
YKA27T-3-0	27	27	1	16777216	23	627	23	1909	23	654817	23	487	23
YKA27T-3-0	27	27	2	16777216	23	577	23	1885	23	1709141	23	545	23
YKA27T-3-0	27	27	3	16777216	23	577	23	1885	23	1753687	23	545	23
YKA31T-1-0	31	16	1	2047	17	275	17	31	17	929	17	31	17
YKA31T-1-0	31	16	2	131071	23	461	23	275	23	68559	23	219	23
YKA31T-1-0	31	16	3	131071	23	461	23	275	23	74679	23	219	23
YKA31T-2-0	31	24	1	131071	27	1807	27	83	27	56723	27	81	27
YKA31T-2-0	31	24	2	16777216	31	6973	31	877	31	11690531	31	641	31
YKA31T-2-0	31	24	3	16777216	31	6973	31	877	31	12909755	31	641	31
YKA31T-3-0	31	31	1	8388607	31	10025	31	261	31	759771	31	259	31
YKA31T-3-0	31	31	2	16777216	41	33437	41	3927	41	16777216	41	3705	41
YKA31T-3-0	31	31	3	16777216	41	33437	41	3927	41	16777216	41	3705	41

Table C-3 Results for the detailed Upper Bound experimentation

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	***n***	***m***	**CT**	**# of Nodes**	**Optimal Node**	**Average CPU Time**	**Max CPU Time**	**# of Nodes**	**Optimal Node**	**Average CPU Time**	**Max CPU Time**	**# of Nodes**	**Optimal Node**	**Average CPU Time**	**Max CPU Time**
BOWMAN8-1-0	8	4	1	1	1	0.0000	0.0000	1	1	0.0000	0.0000	1	1	0.0015	0.0150
BOWMAN8-1-0	8	4	2	3	1	0.0015	0.0150	3	0	0.0000	0.0000	3	1	0.0000	0.0000
BOWMAN8-1-0	8	4	3	5	1	0.0000	0.0000	5	0	0.0016	0.0160	5	1	0.0016	0.0160
BOWMAN8-2-0	8	6	1	5	3	0.0016	0.0160	5	3	0.0000	0.0000	7	3	0.0000	0.0000
BOWMAN8-2-0	8	6	2	7	3	0.0000	0.0000	7	3	0.0000	0.0000	7	3	0.0047	0.0160
BOWMAN8-2-0	8	6	3	9	3	0.0000	0.0000	9	3	0.0031	0.0160	11	3	0.0000	0.0000
BOWMAN8-3-0	8	8	1	11	5	0.0016	0.0160	11	5	0.0000	0.0000	13	5	0.0000	0.0000
BOWMAN8-3-0	8	8	2	15	5	0.0015	0.0150	15	5	0.0031	0.0160	25	5	0.0016	0.0160
BOWMAN8-3-0	8	8	3	21	13	0.0032	0.0160	21	13	0.0015	0.0150	35	13	0.0000	0.0000
BUXEY-1-0	29	15	1	7	1	0.0015	0.0150	7	0	0.0030	0.0150	9	1	0.0062	0.0310
BUXEY-1-0	29	15	2	31	19	0.0091	0.0160	27	19	0.0095	0.0160	61	19	0.0045	0.0150
BUXEY-1-0	29	15	3	329	21	0.0765	0.0780	285	21	0.1448	0.1710	437	21	0.0531	0.0630
BUXEY-2-0	29	22	1	15	7	0.0031	0.0160	15	7	0.0094	0.0160	25	7	0.0046	0.0160
BUXEY-2-0	29	22	2	203	31	0.0640	0.0780	177	31	0.1137	0.1250	2437	31	0.3227	0.3290
BUXEY-2-0	29	22	3	3715	81	1.2940	1.4650	3583	81	2.5990	2.6970	9205	181	1.5681	1.5760
BUXEY-3-0	29	29	1	31	9	0.0141	0.0160	27	9	0.0155	0.0160	39	9	0.0077	0.0160
BUXEY-3-0	29	29	2	397	31	0.1714	0.1720	343	31	0.2902	0.2970	4567	31	0.9026	0.9510
BUXEY-3-0	29	29	3	18025	113	8.5873	8.6370	17935	113	18.0102	20.8450	400325	175	88.9858	90.1030
MERTENS-1-0	7	4	1	5	5	0.0000	0.0000	5	5	0.0015	0.0150	7	5	0.0030	0.0150
MERTENS-1-0	7	4	2	5	5	0.0000	0.0000	5	5	0.0000	0.0000	7	5	0.0048	0.0160
MERTENS-1-0	7	4	3	7	5	0.0000	0.0000	7	5	0.0016	0.0160	7	5	0.0030	0.0150
MERTENS-2-0	7	6	1	9	3	0.0032	0.0160	9	3	0.0000	0.0000	9	3	0.0032	0.0160
MERTENS-2-0	7	6	2	17	3	0.0000	0.0000	17	3	0.0000	0.0000	17	3	0.0000	0.0000
MERTENS-2-0	7	6	3	25	3	0.0000	0.0000	25	3	0.0032	0.0160	31	3	0.0032	0.0160
MERTENS-3-0	7	7	1	9	1	0.0015	0.0150	9	0	0.0000	0.0000	25	1	0.0016	0.0160
MERTENS-3-0	7	7	2	11	1	0.0046	0.0160	11	0	0.0000	0.0000	17	1	0.0015	0.0150

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
MERTENS-3-0	7	7	3	13	1	0.0000	0.0000	13	0	0.0000	0.0000	39	1	0.0000	0.0000
HESKIA-1-0	28	14	1	23	17	0.0031	0.0160	23	17	0.0110	0.0160	1795	17	0.1261	0.1400
HESKIA-1-0	28	14	2	91	21	0.0188	0.0320	53	21	0.0264	0.0320	199	21	0.0172	0.0310
HESKIA-1-0	28	14	3	91	21	0.0170	0.0310	53	21	0.0267	0.0320	199	21	0.0264	0.0320
HESKIA-2-0	28	21	1	43	23	0.0125	0.0160	43	23	0.0235	0.0320	175525	23	19.0169	19.3890
HESKIA-2-0	28	21	2	185	23	0.0563	0.0630	177	23	0.1199	0.1250	739	23	0.1261	0.1400
HESKIA-2-0	28	21	3	185	23	0.0625	0.0630	177	23	0.1183	0.1250	739	23	0.1152	0.1250
HESKIA-3-0	28	28	1	73	35	0.0310	0.0310	69	35	0.0564	0.0630	15909	135	2.9182	2.9390
HESKIA-3-0	28	28	2	2081	53	0.9229	0.9350	1931	39	1.8039	1.8090	16149	61	3.5570	3.5950
HESKIA-3-0	28	28	3	2081	53	0.9244	0.9360	1931	39	1.8086	1.8400	16149	61	3.5474	3.5720
JACKSON-1-0	11	6	1	3	1	0.0032	0.0160	3	0	0.0016	0.0160	3	1	0.0031	0.0310
JACKSON-1-0	11	6	2	15	7	0.0000	0.0000	13	7	0.0031	0.0160	21	7	0.0000	0.0000
JACKSON-1-0	11	6	3	15	7	0.0031	0.0160	13	7	0.0031	0.0160	21	7	0.0000	0.0000
JACKSON-2-0	11	9	1	7	7	0.0000	0.0000	7	7	0.0000	0.0000	15	7	0.0000	0.0000
JACKSON-2-0	11	9	2	23	11	0.0046	0.0160	19	11	0.0047	0.0160	99	11	0.0079	0.0160
JACKSON-2-0	11	9	3	23	11	0.0031	0.0160	19	11	0.0031	0.0160	99	11	0.0045	0.0150
JACKSON-3-0	11	11	1	9	3	0.0000	0.0000	9	3	0.0031	0.0160	25	3	0.0000	0.0000
JACKSON-3-0	11	11	2	43	17	0.0048	0.0160	37	17	0.0062	0.0160	337	17	0.0157	0.0160
JACKSON-3-0	11	11	3	43	17	0.0062	0.0160	37	17	0.0093	0.0160	337	17	0.0155	0.0160
JAESCHKE-1-0	9	5	1	1	1	0.0000	0.0000	1	1	0.0000	0.0000	1	1	0.0031	0.0310
JAESCHKE-1-0	9	5	2	3	1	0.0000	0.0000	3	0	0.0016	0.0160	3	1	0.0000	0.0000
JAESCHKE-1-0	9	5	3	7	1	0.0000	0.0000	7	0	0.0016	0.0160	13	1	0.0031	0.0160
JAESCHKE-2-0	9	7	1	3	3	0.0032	0.0160	3	3	0.0000	0.0000	3	3	0.0000	0.0000
JAESCHKE-2-0	9	7	2	9	3	0.0016	0.0160	9	3	0.0000	0.0000	9	3	0.0000	0.0000
JAESCHKE-2-0	9	7	3	17	3	0.0000	0.0000	17	3	0.0000	0.0000	29	3	0.0000	0.0000
JAESCHKE-3-0	9	9	1	7	5	0.0032	0.0160	7	5	0.0016	0.0160	9	5	0.0016	0.0160

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
MERTENS-3-0	7	7	3	13	1	0.0000	0.0000	13	0	0.0000	0.0000	39	1	0.0000	0.0000
HESKIA-1-0	28	14	1	23	17	0.0031	0.0160	23	17	0.0110	0.0160	1795	17	0.1261	0.1400
HESKIA-1-0	28	14	2	91	21	0.0188	0.0320	53	21	0.0264	0.0320	199	21	0.0172	0.0310
HESKIA-1-0	28	14	3	91	21	0.0170	0.0310	53	21	0.0267	0.0320	199	21	0.0264	0.0320
HESKIA-2-0	28	21	1	43	23	0.0125	0.0160	43	23	0.0235	0.0320	175525	23	19.0169	19.3890
HESKIA-2-0	28	21	2	185	23	0.0563	0.0630	177	23	0.1199	0.1250	739	23	0.1261	0.1400
HESKIA-2-0	28	21	3	185	23	0.0625	0.0630	177	23	0.1183	0.1250	739	23	0.1152	0.1250
HESKIA-3-0	28	28	1	73	35	0.0310	0.0310	69	35	0.0564	0.0630	15909	135	2.9182	2.9390
HESKIA-3-0	28	28	2	2081	53	0.9229	0.9350	1931	39	1.8039	1.8090	16149	61	3.5570	3.5950
HESKIA-3-0	28	28	3	2081	53	0.9244	0.9360	1931	39	1.8086	1.8400	16149	61	3.5474	3.5720
JACKSON-1-0	11	6	1	3	1	0.0032	0.0160	3	0	0.0016	0.0160	3	1	0.0031	0.0310
JACKSON-1-0	11	6	2	15	7	0.0000	0.0000	13	7	0.0031	0.0160	21	7	0.0000	0.0000
JACKSON-1-0	11	6	3	15	7	0.0031	0.0160	13	7	0.0031	0.0160	21	7	0.0000	0.0000
JACKSON-2-0	11	9	1	7	7	0.0000	0.0000	7	7	0.0000	0.0000	15	7	0.0000	0.0000
JACKSON-2-0	11	9	2	23	11	0.0046	0.0160	19	11	0.0047	0.0160	99	11	0.0079	0.0160
JACKSON-2-0	11	9	3	23	11	0.0031	0.0160	19	11	0.0031	0.0160	99	11	0.0045	0.0150
JACKSON-3-0	11	11	1	9	3	0.0000	0.0000	9	3	0.0031	0.0160	25	3	0.0000	0.0000
JACKSON-3-0	11	11	2	43	17	0.0048	0.0160	37	17	0.0062	0.0160	337	17	0.0157	0.0160
JACKSON-3-0	11	11	3	43	17	0.0062	0.0160	37	17	0.0093	0.0160	337	17	0.0155	0.0160
JAESCHKE-1-0	9	5	1	1	1	0.0000	0.0000	1	1	0.0000	0.0000	1	1	0.0031	0.0310
JAESCHKE-1-0	9	5	2	3	1	0.0000	0.0000	3	0	0.0016	0.0160	3	1	0.0000	0.0000
JAESCHKE-1-0	9	5	3	7	1	0.0000	0.0000	7	0	0.0016	0.0160	13	1	0.0031	0.0160
JAESCHKE-2-0	9	7	1	3	3	0.0032	0.0160	3	3	0.0000	0.0000	3	3	0.0000	0.0000
JAESCHKE-2-0	9	7	2	9	3	0.0016	0.0160	9	3	0.0000	0.0000	9	3	0.0000	0.0000
JAESCHKE-2-0	9	7	3	17	3	0.0000	0.0000	17	3	0.0000	0.0000	29	3	0.0000	0.0000
JAESCHKE-3-0	9	9	1	7	5	0.0032	0.0160	7	5	0.0016	0.0160	9	5	0.0016	0.0160

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
JAESCHKE-3-0	9	9	2	19	5	0.0015	0.0150	19	5	0.0046	0.0160	23	5	0.0031	0.0160
JAESCHKE-3-0	9	9	3	31	5	0.0016	0.0160	31	5	0.0000	0.0000	57	5	0.0015	0.0150
LUTZ1-1-0	32	16	1	9	5	0.0031	0.0160	9	5	0.0079	0.0160	11	5	0.0062	0.0310
LUTZ1-1-0	32	16	2	75	19	0.0156	0.0160	53	19	0.0297	0.0320	283	19	0.0296	0.0320
LUTZ1-1-0	32	16	3	165	19	0.0389	0.0470	139	19	0.0688	0.0780	333	19	0.0357	0.0470
LUTZ1-2-0	32	24	1	17	11	0.0047	0.0160	17	11	0.0109	0.0160	17	11	0.0032	0.0160
LUTZ1-2-0	32	24	2	499	11	0.1700	0.1720	349	11	0.2292	0.2340	8615	11	1.2512	1.2780
LUTZ1-2-0	32	24	3	2115	11	0.7997	0.8110	1831	11	1.3952	1.4030	5149	11	0.9629	0.9820
LUTZ1-3-0	32	32	1	65	13	0.0313	0.0320	51	13	0.0374	0.0470	55	13	0.0155	0.0310
LUTZ1-3-0	32	32	2	3929	31	1.9020	1.9020	3767	31	3.4174	3.4300	454341	31	90.8558	91.6150
LUTZ1-3-0	32	32	3	15845	31	8.4563	8.5130	13053	31	13.9973	14.0160	110517	31	28.4940	29.4260
MANSOOR-1-0	11	6	1	1	1	0.0000	0.0000	1	1	0.0015	0.0150	1	1	0.0031	0.0310
MANSOOR-1-0	11	6	2	11	9	0.0016	0.0160	11	9	0.0015	0.0150	13	9	0.0016	0.0160
MANSOOR-1-0	11	6	3	21	9	0.0015	0.0150	17	9	0.0016	0.0160	19	9	0.0000	0.0000
MANSOOR-2-0	11	9	1	5	1	0.0000	0.0000	5	0	0.0016	0.0160	7	1	0.0000	0.0000
MANSOOR-2-0	11	9	2	19	15	0.0032	0.0160	19	15	0.0031	0.0160	21	15	0.0016	0.0160
MANSOOR-2-0	11	9	3	35	15	0.0032	0.0160	27	15	0.0015	0.0150	65	15	0.0032	0.0160
MANSOOR-3-0	11	11	1	11	1	0.0030	0.0150	11	0	0.0016	0.0160	15	1	0.0030	0.0150
MANSOOR-3-0	11	11	2	23	19	0.0016	0.0160	21	19	0.0030	0.0150	113	19	0.0076	0.0160
MANSOOR-3-0	11	11	3	43	19	0.0078	0.0160	43	19	0.0062	0.0160	423	19	0.0174	0.0310
MITCHELL-1-0	21	11	1	7	5	0.0000	0.0000	7	5	0.0063	0.0160	7	5	0.0030	0.0150
MITCHELL-1-0	21	11	2	25	13	0.0032	0.0160	21	13	0.0046	0.0160	51	13	0.0032	0.0160
MITCHELL-1-0	21	11	3	49	13	0.0108	0.0160	45	13	0.0110	0.0160	65	13	0.0076	0.0160
MITCHELL-2-0	21	16	1	9	7	0.0031	0.0160	9	7	0.0016	0.0160	13	7	0.0016	0.0160
MITCHELL-2-0	21	16	2	67	21	0.0124	0.0160	57	21	0.0187	0.0310	433	21	0.0344	0.0470
MITCHELL-2-0	21	16	3	121	25	0.0263	0.0320	117	25	0.0454	0.0470	4013	25	0.3115	0.3130

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
MITCHELL-3-0	21	21	1	49	7	0.0127	0.0160	45	7	0.0123	0.0160	45	7	0.0031	0.0160
MITCHELL-3-0	21	21	2	323	23	0.0749	0.0780	249	23	0.1138	0.1250	1359	23	0.1513	0.1560
MITCHELL-3-0	21	21	3	1777	25	0.4336	0.4370	1679	25	0.8012	0.8110	8517	25	0.9852	0.9980
ROSZIEG-1-0	25	13	1	11	7	0.0015	0.0150	11	7	0.0046	0.0160	15	7	0.0061	0.0160
ROSZIEG-1-0	25	13	2	31	15	0.0062	0.0160	31	15	0.0079	0.0160	83	15	0.0095	0.0160
ROSZIEG-1-0	25	13	3	47	15	0.0111	0.0160	45	15	0.0154	0.0160	51	15	0.0032	0.0160
ROSZIEG-2-0	25	19	1	15	7	0.0032	0.0160	15	7	0.0078	0.0160	31	7	0.0048	0.0160
ROSZIEG-2-0	25	19	2	113	19	0.0264	0.0320	101	19	0.0469	0.0470	243	19	0.0281	0.0320
ROSZIEG-2-0	25	19	3	283	19	0.0701	0.0780	271	19	0.1339	0.1410	681	19	0.0825	0.0940
ROSZIEG-3-0	25	25	1	49	9	0.0141	0.0160	49	9	0.0265	0.0320	113	9	0.0170	0.0310
ROSZIEG-3-0	25	25	2	659	27	0.2137	0.2190	645	27	0.4005	0.4210	3937	27	0.6048	0.6100
ROSZIEG-3-0	25	25	3	2733	27	0.9246	0.9360	2683	27	1.7898	1.7930	14535	27	2.3677	2.4150
SAWYER30-1-0	30	15	1	11	9	0.0016	0.0160	11	9	0.0047	0.0160	31	9	0.0046	0.0310
SAWYER30-1-0	30	15	2	93	21	0.0233	0.0310	49	21	0.0216	0.0320	155	21	0.0189	0.0310
SAWYER30-1-0	30	15	3	183	45	0.0436	0.0470	169	43	0.0906	0.0940	881	97	0.0951	0.1090
SAWYER30-2-0	30	23	1	75	17	0.0184	0.0320	63	17	0.0327	0.0470	153	17	0.0249	0.0320
SAWYER30-2-0	30	23	2	195	25	0.0686	0.0780	145	25	0.0999	0.1090	1169	25	0.1856	0.1880
SAWYER30-2-0	30	23	3	2245	29	0.8465	0.8580	2181	29	1.7212	1.7310	20491	29	3.5788	3.6160
SAWYER30-3-0	30	30	1	235	75	0.0889	0.0940	139	61	0.0967	0.1090	5977	2133	1.0117	1.0600
SAWYER30-3-0	30	30	2	1087	121	0.5241	0.5310	1017	115	0.9945	1.0130	1662885	12535	317.8756	320.9430
SAWYER30-3-0	30	30	3	15835	749	8.1071	8.1230	15753	749	16.6257	16.6660	1226211	31819	283.3909	291.0330
AKKSKT12-1-0	12	6	1	9	9	0.0000	0.0000	9	9	0.0016	0.0160	21	9	0.0016	0.0160
AKKSKT12-1-0	12	6	2	11	9	0.0000	0.0000	11	9	0.0015	0.0150	15	9	0.0000	0.0000
AKKSKT12-1-0	12	6	3	15	9	0.0061	0.0160	13	9	0.0000	0.0000	13	9	0.0016	0.0160
AKKSKT12-2-0	12	9	1	13	11	0.0015	0.0150	13	11	0.0000	0.0000	39	11	0.0031	0.0160
AKKSKT12-2-0	12	9	2	23	11	0.0015	0.0150	17	11	0.0047	0.0160	41	11	0.0032	0.0160

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	***n***	***m***	**CT**	**# of Nodes**	**Optimal Node**	**Average CPU Time**	**Max CPU Time**	**# of Nodes**	**Optimal Node**	**Average CPU Time**	**Max CPU Time**	**# of Nodes**	**Optimal Node**	**Average CPU Time**	**Max CPU Time**
AKKSKT12-2-0	12	9	3	37	11	0.0032	0.0160	19	11	0.0016	0.0160	45	11	0.0016	0.0160
AKKSKT12-3-0	12	12	1	25	11	0.0016	0.0160	17	11	0.0048	0.0160	39	11	0.0031	0.0160
AKKSKT12-3-0	12	12	2	51	11	0.0047	0.0160	33	11	0.0047	0.0160	99	11	0.0063	0.0170
AKKSKT12-3-0	12	12	3	81	15	0.0092	0.0160	37	13	0.0110	0.0160	79	17	0.0047	0.0160
AKO20T-1-0	15	8	1	3	3	0.0000	0.0000	3	3	0.0000	0.0000	3	3	0.0046	0.0310
AKO20T-1-0	15	8	2	15	3	0.0000	0.0000	13	3	0.0016	0.0160	15	3	0.0000	0.0000
AKO20T-1-0	15	8	3	23	3	0.0016	0.0160	19	3	0.0076	0.0160	19	3	0.0016	0.0160
AKO20T-2-0	15	12	1	7	3	0.0000	0.0000	7	3	0.0015	0.0150	9	3	0.0000	0.0000
AKO20T-2-0	15	12	2	83	65	0.0109	0.0160	75	57	0.0158	0.0160	107	83	0.0079	0.0160
AKO20T-2-0	15	12	3	183	129	0.0170	0.0310	145	97	0.0343	0.0470	215	151	0.0125	0.0180
AKO20T-3-0	15	15	1	13	1	0.0047	0.0160	13	0	0.0032	0.0160	49	1	0.0031	0.0160
AKO20T-3-0	15	15	2	145	1	0.0186	0.0310	113	0	0.0296	0.0320	445	1	0.0299	0.0350
AKO20T-3-0	15	15	3	507	71	0.0702	0.0780	383	67	0.1168	0.1250	913	95	0.0693	0.0780
AKO30T1-1-0	16	8	1	5	5	0.0016	0.0160	5	5	0.0015	0.0150	7	5	0.0032	0.0160
AKO30T1-1-0	16	8	2	23	5	0.0031	0.0160	21	5	0.0093	0.0160	29	5	0.0000	0.0000
AKO30T1-1-0	16	8	3	23	5	0.0000	0.0000	21	5	0.0015	0.0150	29	5	0.0015	0.0150
AKO30T1-2-0	16	12	1	17	7	0.0000	0.0000	17	7	0.0095	0.0160	127	7	0.0063	0.0170
AKO30T1-2-0	16	12	2	69	11	0.0032	0.0160	57	11	0.0122	0.0160	301	11	0.0191	0.0310
AKO30T1-2-0	16	12	3	69	11	0.0141	0.0160	57	11	0.0095	0.0160	301	11	0.0157	0.0170
AKO30T1-3-0	16	16	1	67	31	0.0031	0.0160	57	31	0.0123	0.0160	683	51	0.0408	0.0520
AKO30T1-3-0	16	16	2	95	31	0.0159	0.0160	95	31	0.0295	0.0320	2495	31	0.1813	0.1920
AKO30T1-3-0	16	16	3	95	31	0.0156	0.0160	95	31	0.0297	0.0320	2495	31	0.1794	0.1920
AKO30T2-1-0	14	7	1	5	1	0.0016	0.0160	5	0	0.0031	0.0160	5	1	0.0047	0.0310
AKO30T2-1-0	14	7	2	19	1	0.0015	0.0150	17	0	0.0000	0.0000	31	1	0.0018	0.0180
AKO30T2-1-0	14	7	3	19	1	0.0015	0.0150	17	0	0.0015	0.0150	31	1	0.0015	0.0150
AKO30T2-2-0	14	11	1	7	3	0.0016	0.0160	7	3	0.0016	0.0160	9	3	0.0000	0.0000
AKO30T2-2-0	14	11	2	29	13	0.0000	0.0000	23	13	0.0032	0.0160	195	13	0.0110	0.0160
AKO30T2-2-0	14	11	3	29	13	0.0109	0.0160	25	13	0.0000	0.0000	195	13	0.0110	0.0170

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
AKO30T2-3-0	14	14	1	17	3	0.0015	0.0150	13	3	0.0016	0.0160	41	3	0.0031	0.0160
AKO30T2-3-0	14	14	2	59	31	0.0032	0.0160	51	31	0.0158	0.0160	255	99	0.0172	0.0310
AKO30T2-3-0	14	14	3	59	31	0.0063	0.0160	51	31	0.0107	0.0160	255	99	0.0158	0.0170
GUN17T-1-0	17	9	1	7	7	0.0016	0.0160	7	7	0.0064	0.0160	13	7	0.0032	0.0320
GUN17T-1-0	17	9	2	55	7	0.0016	0.0160	27	7	0.0032	0.0160	225	7	0.0097	0.0180
GUN17T-1-0	17	9	3	73	17	0.0109	0.0160	39	17	0.0030	0.0150	341	17	0.0110	0.0170
GUN17T-2-0	17	13	1	15	15	0.0046	0.0160	15	15	0.0016	0.0160	145	15	0.0109	0.0160
GUN17T-2-0	17	13	2	43	15	0.0032	0.0160	25	15	0.0125	0.0160	871	15	0.0472	0.0520
GUN17T-2-0	17	13	3	85	15	0.0155	0.0160	39	15	0.0094	0.0160	2193	15	0.1260	0.1400
GUN17T-3-0	17	17	1	27	19	0.0016	0.0160	27	19	0.0093	0.0160	4095	19	0.2646	0.2970
GUN17T-3-0	17	17	2	75	19	0.0109	0.0160	71	19	0.0250	0.0320	20749	19	1.4855	1.6400
GUN17T-3-0	17	17	3	115	27	0.0250	0.0320	111	27	0.0405	0.0470	41821	365	3.0148	3.3340
GUN8T-1-0	8	4	1	1	1	0.0015	0.0150	1	1	0.0000	0.0000	1	1	0.0015	0.0150
GUN8T-1-0	8	4	2	3	3	0.0000	0.0000	3	3	0.0000	0.0000	3	3	0.0000	0.0000
GUN8T-1-0	8	4	3	9	3	0.0016	0.0160	7	3	0.0000	0.0000	9	3	0.0016	0.0160
GUN8T-2-0	8	6	1	3	3	0.0016	0.0160	3	3	0.0015	0.0150	3	3	0.0016	0.0160
GUN8T-2-0	8	6	2	5	5	0.0015	0.0150	5	5	0.0000	0.0000	7	5	0.0015	0.0150
GUN8T-2-0	8	6	3	9	5	0.0015	0.0150	9	5	0.0016	0.0160	9	5	0.0015	0.0150
GUN8T-3-0	8	8	1	5	3	0.0000	0.0000	5	3	0.0000	0.0000	7	3	0.0000	0.0000
GUN8T-3-0	8	8	2	11	5	0.0000	0.0000	9	5	0.0015	0.0150	17	5	0.0049	0.0180
GUN8T-3-0	8	8	3	29	11	0.0031	0.0160	27	11	0.0048	0.0160	55	11	0.0016	0.0160
LAM20T-1-0	9	5	1	1	1	0.0016	0.0160	1	1	0.0016	0.0160	1	1	0.0016	0.0160
LAM20T-1-0	9	5	2	21	9	0.0016	0.0160	11	9	0.0000	0.0000	31	9	0.0016	0.0160
LAM20T-1-0	9	5	3	21	9	0.0016	0.0160	11	9	0.0000	0.0000	31	9	0.0016	0.0160
LAM20T-2-0	9	7	1	3	3	0.0000	0.0000	3	3	0.0031	0.0160	3	3	0.0032	0.0160
LAM20T-2-0	9	7	2	21	9	0.0015	0.0150	15	9	0.0031	0.0160	21	9	0.0000	0.0000
LAM20T-2-0	9	7	3	21	9	0.0015	0.0150	15	9	0.0000	0.0000	21	9	0.0000	0.0000
LAM20T-3-0	9	9	1	7	3	0.0031	0.0160	7	3	0.0000	0.0000	7	3	0.0000	0.0000

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
LAM20T-3-0	9	9	2	45	3	0.0031	0.0160	31	3	0.0063	0.0160	57	3	0.0032	0.0160
LAM20T-3-0	9	9	3	45	3	0.0000	0.0000	31	3	0.0031	0.0160	57	3	0.0047	0.0170
LAM30T-1-0	9	5	1	1	1	0.0000	0.0000	1	1	0.0015	0.0150	1	1	0.0031	0.0160
LAM30T-1-0	9	5	2	5	1	0.0000	0.0000	5	0	0.0032	0.0160	7	1	0.0000	0.0000
LAM30T-1-0	9	5	3	15	1	0.0015	0.0150	15	0	0.0000	0.0000	17	1	0.0000	0.0000
LAM30T-2-0	9	7	1	3	3	0.0015	0.0150	3	3	0.0000	0.0000	3	3	0.0000	0.0000
LAM30T-2-0	9	7	2	9	3	0.0016	0.0160	9	3	0.0030	0.0150	11	3	0.0015	0.0150
LAM30T-2-0	9	7	3	19	3	0.0016	0.0160	13	3	0.0000	0.0000	15	3	0.0030	0.0150
LAM30T-3-0	9	9	1	7	7	0.0016	0.0160	7	7	0.0016	0.0160	7	7	0.0015	0.0150
LAM30T-3-0	9	9	2	21	7	0.0016	0.0160	21	7	0.0016	0.0160	39	7	0.0000	0.0000
LAM30T-3-0	9	9	3	23	13	0.0016	0.0160	23	13	0.0015	0.0150	37	13	0.0015	0.0150
MAS30T-1-0	30	15	1	19	15	0.0047	0.0160	19	15	0.0063	0.0160	117	15	0.0191	0.0470
MAS30T-1-0	30	15	2	75	15	0.0172	0.0310	29	15	0.0155	0.0160	31	15	0.0016	0.0160
MAS30T-1-0	30	15	3	155	15	0.0373	0.0470	45	15	0.0250	0.0320	45	15	0.0065	0.0170
MAS30T-2-0	30	23	1	107	25	0.0359	0.0470	83	25	0.0530	0.0630	323	25	0.0503	0.0620
MAS30T-2-0	30	23	2	825	179	0.2823	0.2960	455	89	0.3382	0.3430	543	119	0.0977	0.1090
MAS30T-2-0	30	23	3	1699	241	0.6252	0.6400	1137	159	0.9214	0.9360	1625	247	0.3075	0.3320
MAS30T-3-0	30	30	1	385	43	0.1654	0.1870	305	43	0.2555	0.2650	1081	43	0.2225	0.2440
MAS30T-3-0	30	30	2	6233	291	2.9888	2.9940	4367	215	4.2515	4.2870	6873	483	1.6479	1.7970
MAS30T-3-0	30	30	3	22797	351	11.5805	11.6150	19431	259	20.7884	20.8130	30395	547	7.8836	8.7270
MGG7T-1-0	7	4	1	5	5	0.0000	0.0000	5	5	0.0000	0.0000	7	5	0.0016	0.0160
MGG7T-1-0	7	4	2	7	7	0.0000	0.0000	7	7	0.0000	0.0000	9	7	0.0000	0.0000
MGG7T-1-0	7	4	3	7	7	0.0016	0.0160	7	7	0.0000	0.0000	9	7	0.0000	0.0000
MGG7T-2-0	7	6	1	7	7	0.0000	0.0000	7	7	0.0032	0.0160	9	7	0.0030	0.0150
MGG7T-2-0	7	6	2	13	9	0.0000	0.0000	11	9	0.0015	0.0150	13	9	0.0000	0.0000
MGG7T-2-0	7	6	3	13	9	0.0032	0.0160	11	9	0.0000	0.0000	13	9	0.0000	0.0000
MGG7T-3-0	7	7	1	9	7	0.0000	0.0000	9	7	0.0000	0.0000	11	7	0.0032	0.0160
MGG7T-3-0	7	7	2	13	11	0.0000	0.0000	13	11	0.0032	0.0160	25	11	0.0000	0.0000

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
MGG7T-3-0	7	7	3	13	11	0.0046	0.0160	13	11	0.0000	0.0000	25	11	0.0016	0.0160
WANG18T-1-0	20	10	1	11	9	0.0046	0.0160	11	9	0.0000	0.0000	63	9	0.0031	0.0160
WANG18T-1-0	20	10	2	83	9	0.0108	0.0160	31	9	0.0063	0.0160	97	9	0.0095	0.0180
WANG18T-1-0	20	10	3	83	9	0.0109	0.0160	31	9	0.0093	0.0160	97	9	0.0047	0.0160
WANG18T-2-0	20	15	1	19	13	0.0063	0.0160	19	13	0.0016	0.0160	391	13	0.0252	0.0350
WANG18T-2-0	20	15	2	291	13	0.0515	0.0630	157	13	0.0560	0.0630	299	13	0.0237	0.0320
WANG18T-2-0	20	15	3	291	13	0.0501	0.0630	157	13	0.0575	0.0630	299	13	0.0283	0.0350
WANG18T-3-0	20	20	1	91	19	0.0173	0.0310	79	19	0.0280	0.0320	3461	19	0.2871	0.3140
WANG18T-3-0	20	20	2	2329	29	0.5442	0.5460	1429	29	0.6966	0.7020	15315	29	1.6388	1.8150
WANG18T-3-0	20	20	3	2329	29	0.5442	0.5460	1461	29	0.7107	0.7180	15315	29	1.6500	1.8160
YKA19T-1-0	19	10	1	21	13	0.0032	0.0160	17	13	0.0063	0.0160	75	13	0.0093	0.0310
YKA19T-1-0	19	10	2	27	13	0.0031	0.0160	21	13	0.0047	0.0160	55	13	0.0048	0.0160
YKA19T-1-0	19	10	3	27	13	0.0078	0.0160	21	13	0.0031	0.0160	55	13	0.0030	0.0150
YKA19T-2-0	19	15	1	61	13	0.0124	0.0160	59	13	0.0159	0.0160	639	13	0.0491	0.0630
YKA19T-2-0	19	15	2	159	13	0.0279	0.0310	129	13	0.0373	0.0470	1021	13	0.0804	0.0940
YKA19T-2-0	19	15	3	159	13	0.0249	0.0320	129	13	0.0452	0.0470	1023	13	0.0789	0.0870
YKA19T-3-0	19	19	1	159	23	0.0327	0.0470	87	23	0.0374	0.0470	68833	23	5.8157	6.4410
YKA19T-3-0	19	19	2	595	25	0.1311	0.1410	425	25	0.1870	0.1870	114439	25	10.4435	11.5370
YKA19T-3-0	19	19	3	595	25	0.1278	0.1410	425	25	0.1855	0.1880	114567	25	10.4448	11.5540
YKA27T-1-0	27	14	1	545	11	0.1061	0.1100	79	11	0.0313	0.0320	125	11	0.0191	0.0470
YKA27T-1-0	27	14	2	677	11	0.1262	0.1410	173	11	0.0764	0.0780	285	11	0.0265	0.0320
YKA27T-1-0	27	14	3	677	11	0.1327	0.1410	205	11	0.0872	0.0940	285	11	0.0318	0.0350
YKA27T-2-0	27	21	1	287	23	0.0903	0.0940	175	23	0.1123	0.1250	459	23	0.0786	0.0930
YKA27T-2-0	27	21	2	457	23	0.1464	0.1560	357	23	0.2372	0.2500	1529	23	0.2509	0.2800
YKA27T-2-0	27	21	3	457	23	0.1480	0.1560	357	23	0.2387	0.2500	1529	23	0.2494	0.2790
YKA27T-3-0	27	27	1	1909	23	0.7734	0.7800	487	23	0.3803	0.3900	647	23	0.1308	0.1410
YKA27T-3-0	27	27	2	1885	23	0.7782	0.7800	545	23	0.4441	0.4520	595	23	0.1262	0.1390
YKA27T-3-0	27	27	3	1885	23	0.7781	0.7800	545	23	0.4457	0.4520	595	23	0.1261	0.1400

Table C-3 Results for the detailed Upper Bound experimentation (Continued)

				UB 2				UB BEST				BEST OF UB 1 UB 2			
Test Problem	*n*	*m*	CT	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
YKA31T-1-0	31	16	1	31	17	0.0062	0.0160	31	17	0.0125	0.0160	251	17	0.0299	0.0620
YKA31T-1-0	31	16	2	275	23	0.0700	0.0780	219	23	0.1232	0.1250	461	23	0.0602	0.0700
YKA31T-1-0	31	16	3	275	23	0.0749	0.0780	219	23	0.1231	0.1250	461	23	0.0616	0.0690
YKA31T-2-0	31	24	1	83	27	0.0295	0.0320	81	27	0.0609	0.0630	1509	27	0.2507	0.2800
YKA31T-2-0	31	24	2	877	31	0.3618	0.3750	641	31	0.5752	0.5770	6795	31	1.3817	1.5180
YKA31T-2-0	31	24	3	877	31	0.3602	0.3740	641	31	0.5768	0.5770	6795	31	1.3769	1.5190
YKA31T-3-0	31	31	1	261	31	0.1263	0.1400	259	31	0.2323	0.2340	7093	31	1.5963	1.7630
YKA31T-3-0	31	31	2	3927	41	2.1640	2.1670	3705	41	4.3232	4.3490	33269	41	9.2730	10.2450
YKA31T-3-0	31	31	3	3927	41	2.1842	2.2140	3705	41	4.3716	4.6930	33269	41	9.2827	10.2630

Table C-4 Results for the reduction mechanism experimentation

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
BOWMAN8-1-0	8	4	1	1	1	0.0031	0.0160	1	1	0.0000	0.0000
BOWMAN8-1-0	8	4	2	1	1	0.0000	0.0000	3	1	0.0015	0.0150
BOWMAN8-1-0	8	4	3	1	1	0.0000	0.0000	5	1	0.0000	0.0000
BOWMAN8-2-0	8	6	1	3	3	0.0062	0.0160	5	3	0.0016	0.0160
BOWMAN8-2-0	8	6	2	3	3	0.0032	0.0160	7	3	0.0000	0.0000
BOWMAN8-2-0	8	6	3	5	3	0.0000	0.0000	9	3	0.0000	0.0000
BOWMAN8-3-0	8	8	1	7	5	0.0031	0.0160	11	5	0.0016	0.0160
BOWMAN8-3-0	8	8	2	9	5	0.0015	0.0150	15	5	0.0015	0.0150
BOWMAN8-3-0	8	8	3	13	13	0.0045	0.0150	21	13	0.0032	0.0160
BUXEY-1-0	29	15	1	5	0	0.0063	0.0310	7	1	0.0015	0.0150
BUXEY-1-0	29	15	2	27	19	0.0047	0.0160	31	19	0.0091	0.0160
BUXEY-1-0	29	15	3	133	21	0.1094	0.1100	329	21	0.0765	0.0780
BUXEY-2-0	29	22	1	11	5	0.0030	0.0150	15	7	0.0031	0.0160
BUXEY-2-0	29	22	2	71	27	0.0469	0.0470	203	31	0.0640	0.0780
BUXEY-2-0	29	22	3	475	41	0.8717	0.9060	3715	81	1.2940	1.4650
BUXEY-3-0	29	29	1	19	9	0.0079	0.0160	31	9	0.0141	0.0160
BUXEY-3-0	29	29	2	179	27	0.2109	0.2190	397	31	0.1714	0.1720
BUXEY-3-0	29	29	3	1673	31	5.7532	5.7820	18025	113	8.5873	8.6370
MERTENS-1-0	7	4	1	5	5	0.0031	0.0160	5	5	0.0000	0.0000
MERTENS-1-0	7	4	2	5	5	0.0031	0.0160	5	5	0.0000	0.0000
MERTENS-1-0	7	4	3	5	5	0.0000	0.0000	7	5	0.0000	0.0000
MERTENS-2-0	7	6	1	9	3	0.0032	0.0160	9	3	0.0032	0.0160
MERTENS-2-0	7	6	2	9	3	0.0016	0.0160	17	3	0.0000	0.0000
MERTENS-2-0	7	6	3	9	3	0.0016	0.0160	25	3	0.0000	0.0000
MERTENS-3-0	7	7	1	3	0	0.0000	0.0000	9	1	0.0015	0.0150
MERTENS-3-0	7	7	2	3	0	0.0000	0.0000	11	1	0.0046	0.0160
MERTENS-3-0	7	7	3	1	1	0.0015	0.0150	13	1	0.0000	0.0000

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
HESKIA-1-0	28	14	1	17	17	0.0079	0.0310	23	17	0.0031	0.0160
HESKIA-1-0	28	14	2	29	21	0.0109	0.0160	91	21	0.0188	0.0320
HESKIA-1-0	28	14	3	29	21	0.0064	0.0160	91	21	0.0170	0.0310
HESKIA-2-0	28	21	1	23	23	0.0111	0.0160	43	23	0.0125	0.0160
HESKIA-2-0	28	21	2	23	23	0.0142	0.0160	185	23	0.0563	0.0630
HESKIA-2-0	28	21	3	23	23	0.0123	0.0160	185	23	0.0625	0.0630
HESKIA-3-0	28	28	1	29	29	0.0221	0.0320	73	35	0.0310	0.0310
HESKIA-3-0	28	28	2	29	29	0.0421	0.0470	2081	53	0.9229	0.9350
HESKIA-3-0	28	28	3	29	29	0.0423	0.0470	2081	53	0.9244	0.9360
JACKSON-1-0	11	6	1	1	1	0.0015	0.0150	3	1	0.0032	0.0160
JACKSON-1-0	11	6	2	7	5	0.0015	0.0150	15	7	0.0000	0.0000
JACKSON-1-0	11	6	3	7	5	0.0000	0.0000	15	7	0.0031	0.0160
JACKSON-2-0	11	9	1	7	7	0.0032	0.0160	7	7	0.0000	0.0000
JACKSON-2-0	11	9	2	11	11	0.0016	0.0160	23	11	0.0046	0.0160
JACKSON-2-0	11	9	3	11	11	0.0047	0.0160	23	11	0.0031	0.0160
JACKSON-3-0	11	11	1	5	3	0.0016	0.0160	9	3	0.0000	0.0000
JACKSON-3-0	11	11	2	15	15	0.0031	0.0160	43	17	0.0048	0.0160
JACKSON-3-0	11	11	3	15	15	0.0016	0.0160	43	17	0.0062	0.0160
JAESCHKE-1-0	9	5	1	1	1	0.0032	0.0160	1	1	0.0000	0.0000
JAESCHKE-1-0	9	5	2	1	1	0.0015	0.0150	3	1	0.0000	0.0000
JAESCHKE-1-0	9	5	3	5	0	0.0031	0.0160	7	1	0.0000	0.0000
JAESCHKE-2-0	9	7	1	3	3	0.0016	0.0160	3	3	0.0032	0.0160
JAESCHKE-2-0	9	7	2	5	3	0.0000	0.0000	9	3	0.0016	0.0160
JAESCHKE-2-0	9	7	3	13	3	0.0016	0.0160	17	3	0.0000	0.0000
JAESCHKE-3-0	9	9	1	7	5	0.0000	0.0000	7	5	0.0032	0.0160
JAESCHKE-3-0	9	9	2	9	5	0.0000	0.0000	19	5	0.0015	0.0150
JAESCHKE-3-0	9	9	3	25	5	0.0048	0.0160	31	5	0.0016	0.0160
LUTZ1-1-0	32	16	1	5	5	0.0046	0.0310	9	5	0.0031	0.0160

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
LUTZ1-1-0	32	16	2	45	19	0.0144	0.0160	75	19	0.0156	0.0160
LUTZ1-1-0	32	16	3	71	19	0.0253	0.0320	165	19	0.0389	0.0470
LUTZ1-2-0	32	24	1	11	11	0.0062	0.0160	17	11	0.0047	0.0160
LUTZ1-2-0	32	24	2	209	11	0.1110	0.1250	499	11	0.1700	0.1720
LUTZ1-2-0	32	24	3	439	11	0.4220	0.4380	2115	11	0.7997	0.8110
LUTZ1-3-0	32	32	1	43	13	0.0142	0.0320	65	13	0.0313	0.0320
LUTZ1-3-0	32	32	2	159	31	0.1560	0.1720	3929	31	1.9020	1.9020
LUTZ1-3-0	32	32	3	3741	31	7.9128	7.9540	15845	31	8.4563	8.5130
MANSOOR-1-0	11	6	1	1	1	0.0015	0.0150	1	1	0.0000	0.0000
MANSOOR-1-0	11	6	2	7	7	0.0016	0.0160	11	9	0.0016	0.0160
MANSOOR-1-0	11	6	3	9	7	0.0016	0.0160	21	9	0.0015	0.0150
MANSOOR-2-0	11	9	1	3	0	0.0016	0.0160	5	1	0.0000	0.0000
MANSOOR-2-0	11	9	2	17	15	0.0015	0.0150	19	15	0.0032	0.0160
MANSOOR-2-0	11	9	3	17	15	0.0000	0.0000	35	15	0.0032	0.0160
MANSOOR-3-0	11	11	1	7	0	0.0016	0.0160	11	1	0.0030	0.0150
MANSOOR-3-0	11	11	2	19	19	0.0046	0.0160	23	19	0.0016	0.0160
MANSOOR-3-0	11	11	3	19	19	0.0063	0.0160	43	19	0.0078	0.0160
MITCHELL-1-0	21	11	1	5	5	0.0047	0.0160	7	5	0.0000	0.0000
MITCHELL-1-0	21	11	2	17	13	0.0046	0.0160	25	13	0.0032	0.0160
MITCHELL-1-0	21	11	3	17	13	0.0094	0.0160	49	13	0.0108	0.0160
MITCHELL-2-0	21	16	1	5	5	0.0064	0.0160	9	7	0.0031	0.0160
MITCHELL-2-0	21	16	2	35	17	0.0157	0.0160	67	21	0.0124	0.0160
MITCHELL-2-0	21	16	3	49	21	0.0235	0.0310	121	25	0.0263	0.0320
MITCHELL-3-0	21	21	1	19	7	0.0062	0.0160	49	7	0.0127	0.0160
MITCHELL-3-0	21	21	2	129	23	0.0781	0.0790	323	23	0.0749	0.0780
MITCHELL-3-0	21	21	3	129	23	0.1108	0.1250	1777	25	0.4336	0.4370
ROSZIEG-1-0	25	13	1	7	7	0.0047	0.0310	11	7	0.0015	0.0150
ROSZIEG-1-0	25	13	2	15	15	0.0015	0.0150	31	15	0.0062	0.0160

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
ROSZIEG-1-0	25	13	3	15	15	0.0046	0.0160	47	15	0.0111	0.0160
ROSZIEG-2-0	25	19	1	7	7	0.0016	0.0160	15	7	0.0032	0.0160
ROSZIEG-2-0	25	19	2	61	17	0.0280	0.0320	113	19	0.0264	0.0320
ROSZIEG-2-0	25	19	3	61	17	0.0342	0.0470	283	19	0.0701	0.0780
ROSZIEG-3-0	25	25	1	17	9	0.0048	0.0160	49	9	0.0141	0.0160
ROSZIEG-3-0	25	25	2	191	25	0.1610	0.1870	659	27	0.2137	0.2190
ROSZIEG-3-0	25	25	3	191	25	0.2156	0.2340	2733	27	0.9246	0.9360
SAWYER30-1-0	30	15	1	7	7	0.0031	0.0310	11	9	0.0016	0.0160
SAWYER30-1-0	30	15	2	45	17	0.0156	0.0160	93	21	0.0233	0.0310
SAWYER30-1-0	30	15	3	45	17	0.0375	0.0470	183	45	0.0436	0.0470
SAWYER30-2-0	30	23	1	29	15	0.0079	0.0310	75	17	0.0184	0.0320
SAWYER30-2-0	30	23	2	63	21	0.0486	0.0630	195	25	0.0686	0.0780
SAWYER30-2-0	30	23	3	133	25	0.2266	0.2500	2245	29	0.8465	0.8580
SAWYER30-3-0	30	30	1	59	25	0.0217	0.0320	235	75	0.0889	0.0940
SAWYER30-3-0	30	30	2	229	57	0.4159	0.4220	1087	121	0.5241	0.5310
SAWYER30-3-0	30	30	3	1035	129	3.1684	3.2030	15835	749	8.1071	8.1230
AKKSKT12-1-0	12	6	1	9	9	0.0016	0.0160	9	9	0.0000	0.0000
AKKSKT12-1-0	12	6	2	9	9	0.0016	0.0160	11	9	0.0000	0.0000
AKKSKT12-1-0	12	6	3	9	9	0.0000	0.0000	15	9	0.0061	0.0160
AKKSKT12-2-0	12	9	1	11	11	0.0030	0.0150	13	11	0.0015	0.0150
AKKSKT12-2-0	12	9	2	11	11	0.0000	0.0000	23	11	0.0015	0.0150
AKKSKT12-2-0	12	9	3	11	11	0.0031	0.0160	37	11	0.0032	0.0160
AKKSKT12-3-0	12	12	1	11	11	0.0032	0.0160	25	11	0.0016	0.0160
AKKSKT12-3-0	12	12	2	19	11	0.0016	0.0160	51	11	0.0047	0.0160
AKKSKT12-3-0	12	12	3	19	11	0.0077	0.0160	81	15	0.0092	0.0160
AKO20T-1-0	15	8	1	3	3	0.0016	0.0160	3	3	0.0000	0.0000
AKO20T-1-0	15	8	2	9	3	0.0000	0.0000	15	3	0.0000	0.0000
AKO20T-1-0	15	8	3	13	3	0.0063	0.0160	23	3	0.0016	0.0160

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
AKO20T-2-0	15	12	1	3	3	0.0000	0.0000	7	3	0.0000	0.0000
AKO20T-2-0	15	12	2	39	31	0.0094	0.0160	83	65	0.0109	0.0160
AKO20T-2-0	15	12	3	89	65	0.0313	0.0320	183	129	0.0170	0.0310
AKO20T-3-0	15	15	1	3	0	0.0031	0.0160	13	1	0.0047	0.0160
AKO20T-3-0	15	15	2	9	0	0.0000	0.0000	145	1	0.0186	0.0310
AKO20T-3-0	15	15	3	225	41	0.1077	0.1100	507	71	0.0702	0.0780
AKO30T1-1-0	16	8	1	5	5	0.0016	0.0160	5	5	0.0016	0.0160
AKO30T1-1-0	16	8	2	15	5	0.0016	0.0160	23	5	0.0031	0.0160
AKO30T1-1-0	16	8	3	15	5	0.0030	0.0150	23	5	0.0000	0.0000
AKO30T1-2-0	16	12	1	7	7	0.0015	0.0150	17	7	0.0000	0.0000
AKO30T1-2-0	16	12	2	27	17	0.0062	0.0160	69	11	0.0032	0.0160
AKO30T1-2-0	16	12	3	27	17	0.0063	0.0160	69	11	0.0141	0.0160
AKO30T1-3-0	16	16	1	25	23	0.0047	0.0160	67	31	0.0031	0.0160
AKO30T1-3-0	16	16	2	27	21	0.0108	0.0160	95	31	0.0159	0.0160
AKO30T1-3-0	16	16	3	27	21	0.0094	0.0160	95	31	0.0156	0.0160
AKO30T2-1-0	14	7	1	3	0	0.0015	0.0150	5	1	0.0016	0.0160
AKO30T2-1-0	14	7	2	5	0	0.0016	0.0160	19	1	0.0015	0.0150
AKO30T2-1-0	14	7	3	5	0	0.0016	0.0160	19	1	0.0015	0.0150
AKO30T2-2-0	14	11	1	3	3	0.0016	0.0160	7	3	0.0016	0.0160
AKO30T2-2-0	14	11	2	21	13	0.0063	0.0160	29	13	0.0000	0.0000
AKO30T2-2-0	14	11	3	21	13	0.0080	0.0160	29	13	0.0109	0.0160
AKO30T2-3-0	14	14	1	3	3	0.0016	0.0160	17	3	0.0015	0.0150
AKO30T2-3-0	14	14	2	11	11	0.0030	0.0150	59	31	0.0032	0.0160
AKO30T2-3-0	14	14	3	11	11	0.0048	0.0160	59	31	0.0063	0.0160
GUN17T-1-0	17	9	1	7	7	0.0032	0.0160	7	7	0.0016	0.0160
GUN17T-1-0	17	9	2	19	7	0.0063	0.0160	55	7	0.0016	0.0160
GUN17T-1-0	17	9	3	31	15	0.0063	0.0160	73	17	0.0109	0.0160
GUN17T-2-0	17	13	1	13	13	0.0000	0.0000	15	15	0.0046	0.0160

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
GUN17T-2-0	17	13	2	25	13	0.0064	0.0160	43	15	0.0032	0.0160
GUN17T-2-0	17	13	3	33	13	0.0108	0.0160	85	15	0.0155	0.0160
GUN17T-3-0	17	17	1	15	15	0.0030	0.0150	27	19	0.0016	0.0160
GUN17T-3-0	17	17	2	35	15	0.0062	0.0160	75	19	0.0109	0.0160
GUN17T-3-0	17	17	3	67	33	0.0346	0.0470	115	27	0.0250	0.0320
GUN8T-1-0	8	4	1	1	1	0.0031	0.0160	1	1	0.0015	0.0150
GUN8T-1-0	8	4	2	3	3	0.0015	0.0150	3	3	0.0000	0.0000
GUN8T-1-0	8	4	3	5	3	0.0000	0.0000	9	3	0.0016	0.0160
GUN8T-2-0	8	6	1	3	3	0.0016	0.0160	3	3	0.0016	0.0160
GUN8T-2-0	8	6	2	5	5	0.0016	0.0160	5	5	0.0015	0.0150
GUN8T-2-0	8	6	3	7	5	0.0000	0.0000	9	5	0.0015	0.0150
GUN8T-3-0	8	8	1	3	3	0.0000	0.0000	5	3	0.0000	0.0000
GUN8T-3-0	8	8	2	5	5	0.0000	0.0000	11	5	0.0000	0.0000
GUN8T-3-0	8	8	3	13	11	0.0000	0.0000	29	11	0.0031	0.0160
LAM20T-1-0	9	5	1	1	1	0.0048	0.0160	1	1	0.0016	0.0160
LAM20T-1-0	9	5	2	11	9	0.0031	0.0160	21	9	0.0016	0.0160
LAM20T-1-0	9	5	3	11	9	0.0016	0.0160	21	9	0.0016	0.0160
LAM20T-2-0	9	7	1	3	3	0.0000	0.0000	3	3	0.0000	0.0000
LAM20T-2-0	9	7	2	9	7	0.0031	0.0160	21	9	0.0015	0.0150
LAM20T-2-0	9	7	3	9	7	0.0000	0.0000	21	9	0.0015	0.0150
LAM20T-3-0	9	9	1	3	3	0.0048	0.0160	7	3	0.0031	0.0160
LAM20T-3-0	9	9	2	5	3	0.0000	0.0000	45	3	0.0031	0.0160
LAM20T-3-0	9	9	3	5	3	0.0032	0.0160	45	3	0.0000	0.0000
LAM30T-1-0	9	5	1	1	1	0.0048	0.0320	1	1	0.0000	0.0000
LAM30T-1-0	9	5	2	3	0	0.0016	0.0160	5	1	0.0000	0.0000
LAM30T-1-0	9	5	3	13	0	0.0016	0.0160	15	1	0.0015	0.0150
LAM30T-2-0	9	7	1	3	3	0.0016	0.0160	3	3	0.0015	0.0150
LAM30T-2-0	9	7	2	5	3	0.0031	0.0160	9	3	0.0016	0.0160

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
LAM30T-2-0	9	7	3	9	3	0.0000	0.0000	19	3	0.0016	0.0160
LAM30T-3-0	9	9	1	5	5	0.0032	0.0160	7	7	0.0016	0.0160
LAM30T-3-0	9	9	2	9	5	0.0016	0.0160	21	7	0.0016	0.0160
LAM30T-3-0	9	9	3	13	11	0.0032	0.0160	23	13	0.0016	0.0160
MAS30T-1-0	30	15	1	13	13	0.0047	0.0160	19	15	0.0047	0.0160
MAS30T-1-0	30	15	2	31	13	0.0157	0.0160	75	15	0.0172	0.0310
MAS30T-1-0	30	15	3	31	13	0.0154	0.0160	155	15	0.0373	0.0470
MAS30T-2-0	30	23	1	31	21	0.0110	0.0160	107	25	0.0359	0.0470
MAS30T-2-0	30	23	2	173	61	0.1466	0.1560	825	179	0.2823	0.2960
MAS30T-2-0	30	23	3	173	61	0.2468	0.2660	1699	241	0.6252	0.6400
MAS30T-3-0	30	30	1	37	25	0.0171	0.0310	385	43	0.1654	0.1870
MAS30T-3-0	30	30	2	419	61	0.4828	0.4850	6233	291	2.9888	2.9940
MAS30T-3-0	30	30	3	5843	337	13.6235	13.6720	22797	351	11.5805	11.6150
MGG7T-1-0	7	4	1	3	3	0.0030	0.0150	5	5	0.0000	0.0000
MGG7T-1-0	7	4	2	5	5	0.0015	0.0150	7	7	0.0000	0.0000
MGG7T-1-0	7	4	3	5	5	0.0000	0.0000	7	7	0.0016	0.0160
MGG7T-2-0	7	6	1	5	5	0.0016	0.0160	7	7	0.0000	0.0000
MGG7T-2-0	7	6	2	7	7	0.0000	0.0000	13	9	0.0000	0.0000
MGG7T-2-0	7	6	3	7	7	0.0016	0.0160	13	9	0.0032	0.0160
MGG7T-3-0	7	7	1	7	5	0.0000	0.0000	9	7	0.0000	0.0000
MGG7T-3-0	7	7	2	7	7	0.0000	0.0000	13	11	0.0000	0.0000
MGG7T-3-0	7	7	3	7	7	0.0016	0.0160	13	11	0.0046	0.0160
WANG18T-1-0	20	10	1	9	9	0.0032	0.0320	11	9	0.0046	0.0160
WANG18T-1-0	20	10	2	15	9	0.0032	0.0160	83	9	0.0108	0.0160
WANG18T-1-0	20	10	3	15	9	0.0015	0.0150	83	9	0.0109	0.0160
WANG18T-2-0	20	15	1	13	13	0.0031	0.0160	19	13	0.0063	0.0160
WANG18T-2-0	20	15	2	89	13	0.0189	0.0310	291	13	0.0515	0.0630
WANG18T-2-0	20	15	3	89	13	0.0234	0.0320	291	13	0.0501	0.0630

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
WANG18T-3-0	20	20	1	33	17	0.0064	0.0160	91	19	0.0173	0.0310
WANG18T-3-0	20	20	2	179	21	0.1798	0.1880	2329	29	0.5442	0.5460
WANG18T-3-0	20	20	3	179	21	0.1783	0.1880	2329	29	0.5442	0.5460
YKA19T-1-0	19	10	1	13	13	0.0032	0.0160	21	13	0.0032	0.0160
YKA19T-1-0	19	10	2	13	13	0.0062	0.0160	27	13	0.0031	0.0160
YKA19T-1-0	19	10	3	13	13	0.0015	0.0150	27	13	0.0078	0.0160
YKA19T-2-0	19	15	1	15	9	0.0095	0.0160	61	13	0.0124	0.0160
YKA19T-2-0	19	15	2	29	9	0.0155	0.0160	159	13	0.0279	0.0310
YKA19T-2-0	19	15	3	29	9	0.0109	0.0160	159	13	0.0249	0.0320
YKA19T-3-0	19	19	1	27	17	0.0138	0.0160	159	23	0.0327	0.0470
YKA19T-3-0	19	19	2	53	19	0.0421	0.0470	595	25	0.1311	0.1410
YKA19T-3-0	19	19	3	53	19	0.0438	0.0470	595	25	0.1278	0.1410
YKA27T-1-0	27	14	1	23	9	0.0125	0.0310	545	11	0.1061	0.1100
YKA27T-1-0	27	14	2	23	9	0.0139	0.0160	677	11	0.1262	0.1410
YKA27T-1-0	27	14	3	23	9	0.0155	0.0160	677	11	0.1327	0.1410
YKA27T-2-0	27	21	1	25	19	0.0188	0.0320	287	23	0.0903	0.0940
YKA27T-2-0	27	21	2	21	19	0.0156	0.0160	457	23	0.1464	0.1560
YKA27T-2-0	27	21	3	21	19	0.0123	0.0160	457	23	0.1480	0.1560
YKA27T-3-0	27	27	1	21	17	0.0140	0.0160	1909	23	0.7734	0.7800
YKA27T-3-0	27	27	2	21	17	0.0141	0.0160	1885	23	0.7782	0.7800
YKA27T-3-0	27	27	3	21	17	0.0143	0.0160	1885	23	0.7781	0.7800
YKA31T-1-0	31	16	1	15	13	0.0078	0.0310	31	17	0.0062	0.0160
YKA31T-1-0	31	16	2	25	17	0.0280	0.0320	275	23	0.0700	0.0780
YKA31T-1-0	31	16	3	25	17	0.0280	0.0320	275	23	0.0749	0.0780
YKA31T-2-0	31	24	1	21	21	0.0141	0.0160	83	27	0.0295	0.0320
YKA31T-2-0	31	24	2	215	25	0.3830	0.4070	877	31	0.3618	0.3750
YKA31T-2-0	31	24	3	215	25	0.3860	0.4060	877	31	0.3602	0.3740
YKA31T-3-0	31	31	1	97	29	0.1595	0.1870	261	31	0.1263	0.1400

Table C-4 Results for the reduction mechanism experimentation (Continued)

				REDUCTION				NO REDUCTION			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time	# of Nodes	Optimal Node	Average CPU Time	Max CPU Time
YKA31T-3-0	31	31	2	173	35	0.4453	0.4690	3927	41	2.1640	2.1670
YKA31T-3-0	31	31	3	173	35	0.4453	0.4690	3927	41	2.1842	2.2140

Table C-5 Node Number Results for Computational Analysis -*CT* 1

			CYCLE TIME LEVEL 1							
Test Problem	***n***	***m***	**Average # of Nodes**	**Max # of Nodes**	**Stdev # of Nodes**	**Average Optimal Node**	**Max Optimal Node**	**Average Onode/ Node**	**Max Onode/ Node**
BOWMAN8-1	8	4	4.6	7	2.0656	4.4	7	0.9333	1.0000
BOWMAN8-2	8	6	5.8	9	2.3476	5.2	9	0.9206	1.0000
BOWMAN8-3	8	8	7.6	11	1.6465	6.4	11	0.8298	1.0000
BUXEY-1	29	15	9.2	15	3.3267	7.8	15	0.8094	1.0000
BUXEY-2	29	22	15	27	6.0369	9	17	0.6542	1.0000
BUXEY-3	29	29	24.2	43	9.2472	7.4	17	0.3698	0.8095
MERTENS-1	7	4	4.4	7	1.6465	3.8	7	0.8267	1.0000
MERTENS-2	7	6	6.2	13	3.2931	3.2	7	0.5691	1.0000
MERTENS-3	7	7	6.4	11	2.3190	3.8	7	0.5886	1.0000
HESKIA-1	28	14	17.2	21	2.3944	17.2	21	1.0000	1.0000
HESKIA-2	28	21	24	29	3.0185	23.8	27	0.9931	1.0000
HESKIA-3	28	28	29.2	31	1.1353	29.2	31	1.0000	1.0000
JACKSON-1	11	6	5	9	3.1269	4.6	9	0.9200	1.0000
JACKSON-2	11	9	5.6	9	2.3190	4.8	9	0.8514	1.0000
JACKSON-3	11	11	7.2	11	1.7512	4.4	7	0.6240	1.0000
JAESCHKE-1	9	5	3.6	7	2.1187	3.6	7	1.0000	1.0000
JAESCHKE-2	9	7	5.6	7	1.3499	5	7	0.8800	1.0000
JAESCHKE-3	9	9	4.4	7	1.3499	3.6	5	0.8248	1.0000
LUTZ1-1	32	16	10.6	19	4.8808	9.2	17	0.9005	1.0000
LUTZ1-2	32	24	23	45	12.4365	14.6	25	0.6960	1.0000
LUTZ1-3	32	32	42.6	45	1.2649	13.4	17	0.3147	0.3953
MANSOOR-1	11	6	2.4	5	1.6465	2.2	5	0.9333	1.0000
MANSOOR-2	11	9	3.4	5	1.5776	2.8	5	0.8000	1.0000
MANSOOR-3	11	11	6	9	2.5386	5	7	0.8698	1.0000
MITCHELL-1	21	11	7.6	11	2.3190	7.4	11	0.9818	1.0000
MITCHELL-2	21	16	11.8	23	7.4952	7.6	13	0.7252	1.0000
MITCHELL-3	21	21	11.8	19	3.6757	8	13	0.7180	1.0000
ROSZIEG-1	25	13	10.6	15	3.3731	10	13	0.9511	1.0000
ROSZIEG-2	25	19	15.4	27	6.7198	12	21	0.8397	1.0000
ROSZIEG-3	25	25	26	43	9.3927	13.4	21	0.5704	0.8400
SAWYER30-1	30	15	11.2	21	4.5656	8.8	17	0.7764	1.0000
SAWYER30-2	30	23	19.4	45	11.2665	11	27	0.5504	0.9130
SAWYER30-3	30	30	52	87	20.9603	16.4	31	0.3280	0.5385
AKKSKT12-1	12	6	7	9	1.8856	7	9	1.0000	1.0000
AKKSKT12-2	12	9	8.8	11	1.9889	8.8	11	1.0000	1.0000
AKKSKT12-3	12	12	10.6	15	3.6271	9.4	13	0.9110	1.0000
AKO20T-1	15	8	3.2	7	1.9889	3	7	0.9600	1.0000
AKO20T-2	15	12	5	11	2.4944	5	11	1.0000	1.0000
AKO20T-3	15	15	7.8	11	2.3476	6.6	11	0.8263	1.0000
AKO30T1-1	16	8	8.4	11	2.6750	8.2	11	0.9818	1.0000
AKO30T1-2	16	12	12	21	4.9216	11	19	0.9023	1.0000
AKO30T1-3	16	16	16.8	25	5.2026	14	23	0.8523	1.0000
AKO30T2-1	14	7	5.4	13	3.5024	4.4	7	0.8586	1.0000
AKO30T2-2	14	11	4.6	11	3.0984	3.6	11	0.8076	1.0000
AKO30T2-3	14	14	5.2	9	2.3944	4	9	0.8286	1.0000

Table C-5 Node Number Results for Computational Analysis -*CT* 1 (Continued)

			CYCLE TIME LEVEL 1							
Test Problem	*n*	*m*	Average # of Nodes	Max # of Nodes	Stdev # of Nodes	Average Optimal Node	Max Optimal Node	Average Onode/ Node	Max Onode/ Node
GUN17T-1	17	9	10.4	13	2.1187	10.4	13	1.0000	1.0000
GUN17T-2	17	13	14.6	19	2.0656	13.8	17	0.9474	1.0000
GUN17T-3	17	17	18.2	23	2.5298	16.6	19	0.9198	1.0000
GUN8T-1	8	4	2.6	5	1.8379	2.6	5	1.0000	1.0000
GUN8T-2	8	6	3	5	1.6330	3	5	1.0000	1.0000
GUN8T-3	8	8	3	3	0.0000	3	3	1.0000	1.0000
LAM20T-1	9	5	2	3	1.0541	2	3	1.0000	1.0000
LAM20T-2	9	7	3.8	5	1.3984	3.8	5	1.0000	1.0000
LAM20T-3	9	9	4.8	7	1.1353	4.8	7	1.0000	1.0000
LAM30T-1	9	5	3.8	7	2.1499	3.6	7	0.9333	1.0000
LAM30T-2	9	7	5.2	7	1.7512	5.2	7	1.0000	1.0000
LAM30T-3	9	9	5.8	7	1.0328	5.8	7	1.0000	1.0000
MAS30T-1	30	15	16.8	25	4.6619	16.8	25	1.0000	1.0000
MAS30T-2	30	23	26.2	57	11.7832	21	27	0.8824	1.0000
MAS30T-3	30	30	37	47	6.9921	28.4	41	0.7822	0.9310
MGG7T-1	7	4	2.6	5	1.2649	2.4	5	0.9333	1.0000
MGG7T-2	7	6	3.6	7	1.3499	2.8	5	0.8476	1.0000
MGG7T-3	7	7	5.6	9	1.8974	4.4	5	0.8476	1.0000
WANG18T-1	20	10	10.2	15	2.1499	9.6	15	0.9374	1.0000
WANG18T-2	20	15	14.6	19	2.9515	13.4	19	0.9279	1.0000
WANG18T-3	20	20	19.4	33	5.3166	17.2	19	0.9209	1.0000
YKA19T-1	19	10	10.4	23	5.9666	8.4	13	0.8934	1.0000
YKA19T-2	19	15	16.6	23	4.0879	12.2	17	0.7303	0.8947
YKA19T-3	19	19	34	95	23.7627	12.8	23	0.4631	0.6800
YKA27T-1	27	14	20	33	6.0553	14.2	17	0.7599	1.0000
YKA27T-2	27	21	29.2	51	9.5429	18.4	23	0.6721	0.9200
YKA27T-3	27	27	35.4	67	13.0571	25.6	35	0.7783	1.0000
YKA31T-1	31	16	16.8	27	5.1164	11.4	15	0.7185	1.0000
YKA31T-2	31	24	41.8	77	19.7810	21	47	0.5682	1.0000
YKA31T-3	31	31	104.2	385	112.4404	24	29	0.4065	0.6970

Table C-6 Node Number Results for Computational Analysis - *CT* 2

			CYCLE TIME LEVEL 2							
Test Problem	*n*	*m*	Average # of Nodes	Max # of Nodes	Stdev # of Nodes	Average Optimal Node	Max Optimal Node	Average Onode/ Node	Max Onode/ Node
BOWMAN8-1	8	4	5	7	2.3094	4.8	7	0.9333	1.0000
BOWMAN8-2	8	6	6.8	11	2.5734	6.2	11	0.9206	1.0000
BOWMAN8-3	8	8	10	11	1.4142	8.8	11	0.8747	1.0000
BUXEY-1	29	15	27.6	49	9.6171	16.6	21	0.6355	1.0000
BUXEY-2	29	22	62.4	81	12.8944	25.4	29	0.4203	0.6279
BUXEY-3	29	29	189.4	279	55.3036	36.8	53	0.2052	0.2865
MERTENS-1	7	4	5	7	1.6330	4.4	7	0.8648	1.0000
MERTENS-2	7	6	7.2	13	3.5839	3.4	9	0.5296	1.0000
MERTENS-3	7	7	8.6	17	4.9710	4.6	9	0.5543	1.0000
HESKIA-1	28	14	20.2	29	4.0222	18	23	0.8999	1.0000
HESKIA-2	28	21	23.6	27	2.5033	23.6	27	1.0000	1.0000
HESKIA-3	28	28	126.4	317	128.0279	35.8	49	0.6675	1.0000
JACKSON-1	11	6	8.8	11	1.9889	7.8	11	0.8447	1.0000
JACKSON-2	11	9	15.2	21	3.1903	11.2	17	0.7444	1.0000
JACKSON-3	11	11	19.4	35	6.0955	15	19	0.8312	1.0000
JAESCHKE-1	9	5	5.6	9	2.6750	5.2	9	0.8667	1.0000
JAESCHKE-2	9	7	7.8	9	1.6865	7.2	9	0.9156	1.0000
JAESCHKE-3	9	9	8.6	13	2.4585	5.6	11	0.6356	0.8462
LUTZ1-1	32	16	35.2	89	21.3843	16	23	0.5492	1.0000
LUTZ1-2	32	24	120.6	303	107.0941	23.2	29	0.4102	0.8065
LUTZ1-3	32	32	167.6	197	18.5724	29.6	33	0.1804	0.2441
MANSOOR-1	11	6	7.2	9	1.7512	7.2	9	1.0000	1.0000
MANSOOR-2	11	9	12.4	17	2.9889	12.2	15	0.9882	1.0000
MANSOOR-3	11	11	17	19	1.6330	17	19	1.0000	1.0000
MITCHELL-1	21	11	16.6	25	5.1467	13.4	17	0.8335	1.0000
MITCHELL-2	21	16	31	63	17.0750	18.8	25	0.7236	1.0000
MITCHELL-3	21	21	73.6	129	30.2442	23.2	35	0.3693	0.6571
ROSZIEG-1	25	13	18	29	6.2716	14.4	17	0.8624	1.0000
ROSZIEG-2	25	19	67.6	173	50.4826	15	27	0.3521	0.9310
ROSZIEG-3	25	25	237.6	375	77.3164	24.2	31	0.1149	0.1938
SAWYER30-1	30	15	34	65	12.9701	18	27	0.5937	0.8519
SAWYER30-2	30	23	95.6	183	50.2531	27.6	77	0.3411	0.7196
SAWYER30-3	30	30	432.2	977	256.7856	43.2	63	0.1318	0.2489
AKKSKT12-1	12	6	7.6	9	1.8974	7	9	0.9333	1.0000
AKKSKT12-2	12	9	11.8	27	5.8271	9	11	0.8559	1.0000
AKKSKT12-3	12	12	16.4	25	4.9933	11.8	19	0.7292	1.0000
AKO20T-1	15	8	7.8	11	3.0111	6.2	11	0.7285	1.0000
AKO20T-2	15	12	20.8	45	12.3450	13	31	0.6907	1.0000
AKO20T-3	15	15	21.2	57	15.0982	15.6	47	0.7368	1.0000
AKO30T1-1	16	8	12.2	15	2.1499	11.2	15	0.9333	1.0000
AKO30T1-2	16	12	21.6	27	2.9889	18.2	21	0.8534	1.0000
AKO30T1-3	16	16	32.6	45	7.8202	22.4	27	0.7169	0.9130
AKO30T2-1	14	7	8.8	13	2.5734	8	11	0.8864	1.0000
AKO30T2-2	14	11	20.8	29	4.8488	16.8	25	0.8195	1.0000
AKO30T2-3	14	14	21.2	39	10.0863	17.4	35	0.8672	1.0000

Table C-6 Node Number Results for Computational Analysis - *CT* 2 (Continued)

			CYCLE TIME LEVEL 2							
Test Problem	*n*	*m*	Average # of Nodes	Max # of Nodes	Stdev # of Nodes	Average Optimal Node	Max Optimal Node	Average Onode/ Node	Max Onode/ Node
GUN17T-1	17	9	12.6	19	2.7968	10.8	13	0.8880	1.0000
GUN17T-2	17	13	26.8	55	12.5592	15.6	25	0.6710	1.0000
GUN17T-3	17	17	31.8	79	23.9852	18.2	25	0.7483	1.0000
GUN8T-1	8	4	4.2	7	2.5298	4	7	0.9714	1.0000
GUN8T-2	8	6	4.4	7	1.8974	4	7	0.8933	1.0000
GUN8T-3	8	8	5	5	0.0000	5	5	1.0000	1.0000
LAM20T-1	9	5	8	11	1.9437	6.8	9	0.8379	1.0000
LAM20T-2	9	7	12	15	3.4319	10	15	0.8225	1.0000
LAM20T-3	9	9	7.2	17	3.8239	6	15	0.8352	1.0000
LAM30T-1	9	5	5.6	9	2.1187	4.6	7	0.7492	1.0000
LAM30T-2	9	7	8.6	13	2.4585	7.2	11	0.8537	1.0000
LAM30T-3	9	9	10.6	13	1.2649	8.2	11	0.7765	1.0000
MAS30T-1	30	15	43.6	85	27.2119	19.6	27	0.6056	1.0000
MAS30T-2	30	23	165.4	295	82.9715	36.6	61	0.2614	0.3766
MAS30T-3	30	30	491.4	737	167.4085	47.8	69	0.1068	0.1942
MGG7T-1	7	4	4.8	7	2.2010	4.2	7	0.9143	1.0000
MGG7T-2	7	6	6	11	2.8674	4.6	9	0.8416	1.0000
MGG7T-3	7	7	6.6	9	1.5776	6	9	0.8914	1.0000
WANG18T-1	20	10	15.4	23	4.1952	10.6	15	0.7133	1.0000
WANG18T-2	20	15	38.8	89	26.1865	14	19	0.4954	0.8824
WANG18T-3	20	20	136.2	475	132.5081	17.6	21	0.3039	0.8947
YKA19T-1	19	10	13	31	7.4833	8.4	13	0.7459	1.0000
YKA19T-2	19	15	23.6	57	13.5335	13.2	19	0.6589	1.0000
YKA19T-3	19	19	49.8	121	36.7841	12.8	21	0.3757	0.6800
YKA27T-1	27	14	19.8	33	6.2681	14.2	17	0.7732	1.0000
YKA27T-2	27	21	27	51	9.8432	18.4	23	0.7332	0.9200
YKA27T-3	27	27	32.2	45	6.9410	25.6	35	0.8179	1.0000
YKA31T-1	31	16	62.6	153	47.6823	16.2	27	0.3702	0.7333
YKA31T-2	31	24	230	657	197.2731	25	49	0.2745	0.7576
YKA31T-3	31	31	397.8	1937	567.1286	27.2	35	0.2091	0.6279

Table C-7 Node Number Results for Computational Analysis - *CT* 3

			CYCLE TIME LEVEL 3							
Test Problem	*n*	*m*	Average # of Nodes	Max # of Nodes	Stdev # of Nodes	Average Optimal Node	Max Optimal Node	Average Onode/ Node	Max Onode/ Node
BOWMAN8-1	8	4	5	7	2.3094	4.8	7	0.9333	1.0000
BOWMAN8-2	8	6	7.8	11	2.1499	6.2	11	0.7757	1.0000
BOWMAN8-3	8	8	12.4	15	2.3190	11.6	13	0.9446	1.0000
BUXEY-1	29	15	85.2	147	34.4667	19	25	0.2636	0.4222
BUXEY-2	29	22	381.8	647	168.7475	38.6	69	0.1215	0.2522
BUXEY-3	29	29	2982.6	5393	1449.5976	223.2	963	0.0729	0.2570
MERTENS-1	7	4	5.2	7	1.4757	4.4	7	0.8248	1.0000
MERTENS-2	7	6	7.8	15	4.0222	3.4	9	0.5138	1.0000
MERTENS-3	7	7	9.4	19	5.5618	4.4	9	0.5137	1.0000
HESKIA-1	28	14	20.2	29	4.0222	18	23	0.8999	1.0000
HESKIA-2	28	21	23.6	27	2.5033	23.6	27	1.0000	1.0000
HESKIA-3	28	28	126.4	317	128.0279	35.8	49	0.6675	1.0000
JACKSON-1	11	6	8.8	11	1.9889	7.8	11	0.8447	1.0000
JACKSON-2	11	9	15.2	21	3.1903	11.2	17	0.7444	1.0000
JACKSON-3	11	11	19.4	35	6.0955	15	19	0.8312	1.0000
JAESCHKE-1	9	5	7.6	13	2.5033	6	9	0.8003	1.0000
JAESCHKE-2	9	7	10.8	13	1.9889	7.8	13	0.7568	1.0000
JAESCHKE-3	9	9	22.4	33	7.0585	6.8	13	0.3637	0.8667
LUTZ1-1	32	16	53.4	89	26.5966	16.2	23	0.4245	1.0000
LUTZ1-2	32	24	349	961	308.2784	26.6	39	0.2366	0.7436
LUTZ1-3	32	32	3579.2	4207	769.9422	34.2	51	0.0103	0.0244
MANSOOR-1	11	6	7.4	9	1.8379	7.2	9	0.9778	1.0000
MANSOOR-2	11	9	12.4	17	2.9889	12.2	15	0.9882	1.0000
MANSOOR-3	11	11	17	19	1.6330	17	19	1.0000	1.0000
MITCHELL-1	21	11	18.4	35	8.0028	13.8	17	0.8042	1.0000
MITCHELL-2	21	16	50.6	133	38.1203	19	25	0.5520	1.0000
MITCHELL-3	21	21	163.4	567	149.1004	43.4	221	0.2649	0.6571
ROSZIEG-1	25	13	19.6	29	5.7388	15	17	0.8224	1.0000
ROSZIEG-2	25	19	121	297	87.8736	15	27	0.2674	0.9310
ROSZIEG-3	25	25	455.4	809	197.2968	25	37	0.0618	0.1309
SAWYER30-1	30	15	51.2	111	33.9601	19.8	27	0.5428	0.8519
SAWYER30-2	30	23	246.2	441	141.5476	37	143	0.2320	0.7143
SAWYER30-3	30	30	2219.2	6531	1844.7758	168.8	441	0.1328	0.3437
AKKSKT12-1	12	6	7.6	9	1.8974	7	9	0.9333	1.0000
AKKSKT12-2	12	9	11.8	27	5.8271	9	11	0.8559	1.0000
AKKSKT12-3	12	12	15.4	25	4.6952	10.8	17	0.7199	1.0000
AKO20T-1	15	8	12.6	19	4.2999	6.8	13	0.6077	1.0000
AKO20T-2	15	12	42.2	89	22.9046	20.6	65	0.4960	0.8824
AKO20T-3	15	15	113.8	233	81.8193	39.6	163	0.4195	0.7895
AKO30T1-1	16	8	12.2	15	2.1499	11.2	15	0.9333	1.0000
AKO30T1-2	16	12	21.6	27	2.9889	18.2	21	0.8534	1.0000
AKO30T1-3	16	16	32.6	45	7.8202	22.4	27	0.7169	0.9130
AKO30T2-1	14	7	8.8	13	2.5734	8	11	0.8864	1.0000
AKO30T2-2	14	11	20.8	29	4.8488	16.8	25	0.8195	1.0000
AKO30T2-3	14	14	21.2	39	10.0863	17.4	35	0.8672	1.0000

Table C-7 Node Number Results for Computational Analysis - *CT* 3 (Continued)

			CYCLE TIME LEVEL 3							
Test Problem	*n*	*m*	Average # of Nodes	Max # of Nodes	Stdev # of Nodes	Average Optimal Node	Max Optimal Node	Average Onode/ Node	Max Onode/ Node
GUN17T-1	17	9	19.2	47	11.7927	11.6	15	0.7474	1.0000
GUN17T-2	17	13	46.2	111	32.6490	19.8	39	0.5748	1.0000
GUN17T-3	17	17	66.4	155	45.1865	23.6	33	0.5287	1.0000
GUN8T-1	8	4	4.6	7	2.2706	4	7	0.8648	1.0000
GUN8T-2	8	6	6.6	11	2.6331	6.4	11	0.9714	1.0000
GUN8T-3	8	8	11.4	15	2.6331	8.2	13	0.7017	0.8667
LAM20T-1	9	5	8	11	1.9437	6.8	9	0.8379	1.0000
LAM20T-2	9	7	12	15	3.4319	10	15	0.8225	1.0000
LAM20T-3	9	9	7.2	17	3.8239	6	15	0.8352	1.0000
LAM30T-1	9	5	7.8	13	3.0111	5.8	9	0.8261	1.0000
LAM30T-2	9	7	9.2	13	1.7512	8	11	0.8775	1.0000
LAM30T-3	9	9	15.4	25	4.2999	10.6	13	0.7418	1.0000
MAS30T-1	30	15	95.2	225	80.6058	36.8	113	0.4892	1.0000
MAS30T-2	30	23	578.4	1819	680.6174	61.2	183	0.2212	0.3766
MAS30T-3	30	30	8021	19615	4775.3222	226	435	0.0333	0.0658
MGG7T-1	7	4	4.8	7	2.2010	4.2	7	0.9143	1.0000
MGG7T-2	7	6	6	11	2.8674	4.6	9	0.8416	1.0000
MGG7T-3	7	7	6.6	9	1.5776	6	9	0.8914	1.0000
WANG18T-1	20	10	15.4	23	4.1952	10.6	15	0.7133	1.0000
WANG18T-2	20	15	38.8	89	26.1865	14	19	0.4954	0.8824
WANG18T-3	20	20	136.2	475	132.5081	17.6	21	0.3039	0.8947
YKA19T-1	19	10	13	31	7.4833	8.4	13	0.7459	1.0000
YKA19T-2	19	15	23.6	57	13.5335	13.2	19	0.6589	1.0000
YKA19T-3	19	19	49.8	121	36.7841	12.8	21	0.3757	0.6800
YKA27T-1	27	14	19.8	33	6.2681	14.2	17	0.7732	1.0000
YKA27T-2	27	21	27	51	9.8432	18.4	23	0.7332	0.9200
YKA27T-3	27	27	32.2	45	6.9410	25.6	35	0.8179	1.0000
YKA31T-1	31	16	62.6	153	47.6823	16.2	27	0.3702	0.7333
YKA31T-2	31	24	230	657	197.2731	25	49	0.2745	0.7576
YKA31T-3	31	31	397.8	1937	567.1286	27.2	35	0.2091	0.6279

Table C-8 CPU Time Results for Computational Analysis

			CYCLE TIME LEVEL 1			CYCLE TIME LEVEL 2			CYCLE TIME LEVEL 3		
Test Problem	***n***	***m***	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**
BOWMAN8-1	8	4	0.0020	0.0015	0.0160	0.0006	0.0015	0.0160	0.0009	0.0011	0.0160
BOWMAN8-2	8	6	0.0020	0.0024	0.0160	0.0014	0.0016	0.0160	0.0014	0.0017	0.0160
BOWMAN8-3	8	8	0.0016	0.0015	0.0160	0.0013	0.0010	0.0160	0.0019	0.0022	0.0160
BUXEY-1	29	15	0.0033	0.0024	0.0310	0.0086	0.0048	0.0160	0.0744	0.0325	0.1410
BUXEY-2	29	22	0.0037	0.0019	0.0160	0.0352	0.0093	0.0630	0.6587	0.3115	1.1410
BUXEY-3	29	29	0.0071	0.0024	0.0320	0.1989	0.0651	0.3130	9.4748	4.5322	16.9540
MERTENS-1	7	4	0.0014	0.0014	0.0160	0.0008	0.0011	0.0160	0.0006	0.0011	0.0160
MERTENS-2	7	6	0.0019	0.0010	0.0160	0.0008	0.0008	0.0160	0.0013	0.0012	0.0160
MERTENS-3	7	7	0.0013	0.0012	0.0160	0.0013	0.0014	0.0160	0.0014	0.0016	0.0160
HESKIA-1	28	14	0.0053	0.0018	0.0310	0.0068	0.0037	0.0160	0.0073	0.0036	0.0160
HESKIA-2	28	21	0.0098	0.0041	0.0320	0.0162	0.0030	0.0320	0.0155	0.0037	0.0320
HESKIA-3	28	28	0.0212	0.0025	0.0320	0.1678	0.1752	0.4690	0.1676	0.1723	0.4540
JACKSON-1	11	6	0.0015	0.0013	0.0160	0.0016	0.0011	0.0160	0.0009	0.0008	0.0160
JACKSON-2	11	9	0.0014	0.0017	0.0160	0.0037	0.0022	0.0160	0.0025	0.0020	0.0160
JACKSON-3	11	11	0.0020	0.0017	0.0160	0.0050	0.0017	0.0320	0.0034	0.0026	0.0160
JAESCHKE-1	9	5	0.0022	0.0022	0.0160	0.0005	0.0007	0.0160	0.0019	0.0018	0.0160
JAESCHKE-2	9	7	0.0015	0.0013	0.0160	0.0011	0.0013	0.0160	0.0016	0.0013	0.0160
JAESCHKE-3	9	9	0.0022	0.0011	0.0160	0.0009	0.0008	0.0160	0.0028	0.0027	0.0160
LUTZ1-1	32	16	0.0044	0.0021	0.0310	0.0105	0.0079	0.0320	0.0216	0.0105	0.0470
LUTZ1-2	32	24	0.0078	0.0034	0.0310	0.0641	0.0574	0.2030	0.3463	0.3053	0.9370
LUTZ1-3	32	32	0.0134	0.0017	0.0320	0.1684	0.0258	0.2190	7.6128	1.7343	9.3750
MANSOOR-1	11	6	0.0020	0.0010	0.0160	0.0017	0.0014	0.0160	0.0011	0.0013	0.0160
MANSOOR-2	11	9	0.0017	0.0020	0.0160	0.0028	0.0015	0.0160	0.0003	0.0007	0.0160
MANSOOR-3	11	11	0.0013	0.0010	0.0160	0.0034	0.0025	0.0160	0.0034	0.0031	0.0160
MITCHELL-1	21	11	0.0026	0.0018	0.0160	0.0050	0.0028	0.0160	0.0067	0.0053	0.0160

Table C-8 CPU Time Results for Computational Analysis (Continued)

			CYCLE TIME LEVEL 1			CYCLE TIME LEVEL 2			CYCLE TIME LEVEL 3		
Test Problem	***n***	***m***	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**
MITCHELL-2	21	16	0.0027	0.0021	0.0160	0.0110	0.0066	0.0320	0.0248	0.0172	0.0630
MITCHELL-3	21	21	0.0037	0.0020	0.0160	0.0383	0.0191	0.0790	0.1325	0.1316	0.5000
ROSZIEG-1	25	13	0.0030	0.0020	0.0310	0.0044	0.0032	0.0160	0.0065	0.0026	0.0160
ROSZIEG-2	25	19	0.0038	0.0025	0.0160	0.0346	0.0313	0.1100	0.0851	0.0666	0.2340
ROSZIEG-3	25	25	0.0080	0.0030	0.0310	0.2015	0.0724	0.3600	0.5170	0.2154	0.9220
SAWYER30-1	30	15	0.0038	0.0015	0.0310	0.0133	0.0040	0.0320	0.0380	0.0269	0.0940
SAWYER30-2	30	23	0.0060	0.0046	0.0310	0.0974	0.0626	0.2500	0.4326	0.2724	0.8910
SAWYER30-3	30	30	0.0244	0.0134	0.0780	0.8364	0.6025	2.1570	7.3535	6.5781	22.6880
AKKSKT12-1	12	6	0.0023	0.0018	0.0160	0.0014	0.0019	0.0160	0.0011	0.0011	0.0160
AKKSKT12-2	12	9	0.0021	0.0015	0.0160	0.0014	0.0015	0.0160	0.0023	0.0015	0.0160
AKKSKT12-3	12	12	0.0025	0.0011	0.0160	0.0025	0.0017	0.0160	0.0034	0.0023	0.0160
AKO20T-1	15	8	0.0015	0.0013	0.0160	0.0013	0.0013	0.0160	0.0042	0.0022	0.0160
AKO20T-2	15	12	0.0027	0.0026	0.0160	0.0058	0.0036	0.0160	0.0141	0.0079	0.0320
AKO20T-3	15	15	0.0009	0.0011	0.0160	0.0067	0.0047	0.0160	0.0561	0.0415	0.1250
AKO30T1-1	16	8	0.0023	0.0015	0.0310	0.0020	0.0015	0.0160	0.0017	0.0016	0.0160
AKO30T1-2	16	12	0.0030	0.0014	0.0160	0.0046	0.0029	0.0160	0.0050	0.0027	0.0160
AKO30T1-3	16	16	0.0042	0.0022	0.0160	0.0118	0.0026	0.0310	0.0111	0.0033	0.0160
AKO30T2-1	14	7	0.0017	0.0015	0.0160	0.0012	0.0012	0.0160	0.0012	0.0012	0.0160
AKO30T2-2	14	11	0.0030	0.0016	0.0160	0.0036	0.0021	0.0160	0.0050	0.0019	0.0160
AKO30T2-3	14	14	0.0020	0.0015	0.0160	0.0058	0.0028	0.0160	0.0067	0.0034	0.0160
GUN17T-1	17	9	0.0029	0.0013	0.0160	0.0020	0.0017	0.0160	0.0046	0.0029	0.0160
GUN17T-2	17	13	0.0031	0.0023	0.0160	0.0076	0.0036	0.0160	0.0163	0.0128	0.0470
GUN17T-3	17	17	0.0053	0.0026	0.0160	0.0125	0.0108	0.0320	0.0368	0.0286	0.1090
GUN8T-1	8	4	0.0008	0.0011	0.0160	0.0009	0.0008	0.0160	0.0008	0.0011	0.0160
GUN8T-2	8	6	0.0005	0.0008	0.0160	0.0010	0.0013	0.0160	0.0006	0.0015	0.0160
GUN8T-3	8	8	0.0011	0.0010	0.0160	0.0006	0.0013	0.0160	0.0006	0.0011	0.0160

Table C-8 CPU Time Results for Computational Analysis (Continued)

			CYCLE TIME LEVEL 1			CYCLE TIME LEVEL 2			CYCLE TIME LEVEL 3		
Test Problem	***n***	***m***	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**	**Average CPU Time**	**Stdev CPU Time**	**Max CPU Time**
LAM20T-1	9	5	0.0012	0.0015	0.0160	0.0017	0.0012	0.0160	0.0016	0.0015	0.0160
LAM20T-2	9	7	0.0009	0.0011	0.0160	0.0027	0.0022	0.0160	0.0014	0.0014	0.0160
LAM20T-3	9	9	0.0019	0.0018	0.0160	0.0008	0.0011	0.0160	0.0013	0.0015	0.0160
LAM30T-1	9	5	0.0023	0.0013	0.0320	0.0008	0.0011	0.0160	0.0013	0.0018	0.0160
LAM30T-2	9	7	0.0020	0.0015	0.0160	0.0015	0.0010	0.0160	0.0014	0.0014	0.0160
LAM30T-3	9	9	0.0022	0.0013	0.0160	0.0017	0.0017	0.0320	0.0013	0.0013	0.0160
MAS30T-1	30	15	0.0059	0.0022	0.0160	0.0197	0.0129	0.0470	0.0695	0.0731	0.2350
MAS30T-2	30	23	0.0111	0.0082	0.0470	0.1302	0.0911	0.3750	0.9863	1.4904	4.7030
MAS30T-3	30	30	0.0189	0.0034	0.0320	0.5796	0.1976	0.8750	21.8803	16.0894	60.4540
MGG7T-1	7	4	0.0025	0.0011	0.0160	0.0009	0.0008	0.0160	0.0003	0.0007	0.0160
MGG7T-2	7	6	0.0009	0.0011	0.0160	0.0014	0.0012	0.0160	0.0011	0.0013	0.0160
MGG7T-3	7	7	0.0018	0.0012	0.0160	0.0009	0.0013	0.0160	0.0008	0.0011	0.0160
WANG18T-1	20	10	0.0025	0.0017	0.0320	0.0026	0.0019	0.0160	0.0036	0.0021	0.0160
WANG18T-2	20	15	0.0041	0.0025	0.0160	0.0152	0.0132	0.0470	0.0157	0.0121	0.0470
WANG18T-3	20	20	0.0054	0.0028	0.0160	0.1246	0.1204	0.4070	0.1234	0.1200	0.4060
YKA19T-1	19	10	0.0035	0.0025	0.0160	0.0045	0.0027	0.0160	0.0040	0.0025	0.0160
YKA19T-2	19	15	0.0058	0.0018	0.0320	0.0108	0.0059	0.0320	0.0101	0.0057	0.0320
YKA19T-3	19	19	0.0162	0.0114	0.0470	0.0353	0.0305	0.1100	0.0355	0.0311	0.1100
YKA27T-1	27	14	0.0097	0.0040	0.0320	0.0105	0.0046	0.0160	0.0109	0.0043	0.0320
YKA27T-2	27	21	0.0227	0.0097	0.0470	0.0224	0.0088	0.0470	0.0217	0.0092	0.0470
YKA27T-3	27	27	0.0371	0.0199	0.0940	0.0365	0.0147	0.0780	0.0340	0.0154	0.0790
YKA31T-1	31	16	0.0069	0.0038	0.0310	0.0561	0.0355	0.1560	0.0562	0.0349	0.1560
YKA31T-2	31	24	0.0321	0.0152	0.0630	0.3862	0.3121	1.0630	0.3852	0.3093	1.0470
YKA31T-3	31	31	0.1447	0.1730	0.5780	1.0360	1.6451	5.6250	1.0327	1.6389	5.5630

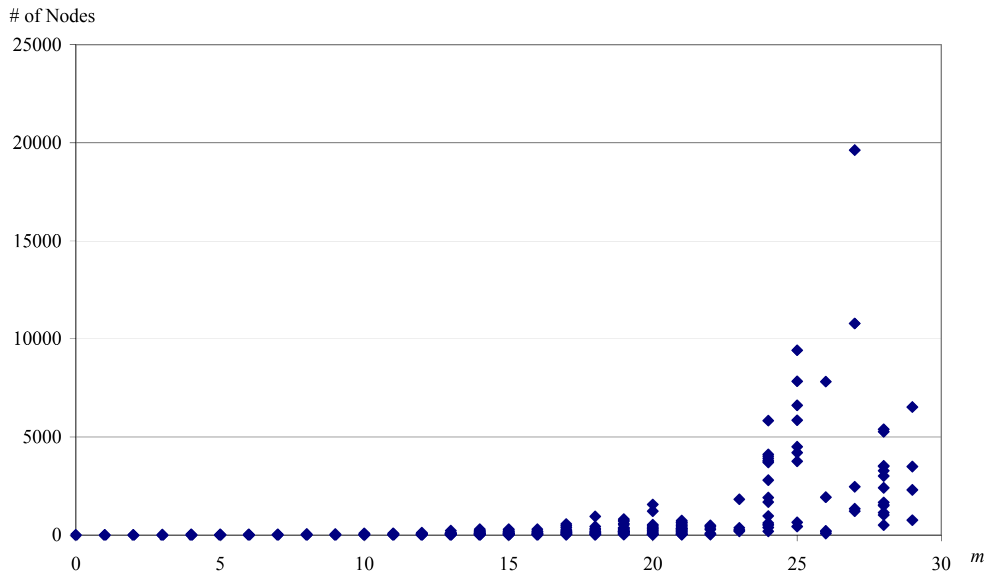


Figure C.1 Number of Nodes versus Number of Parts

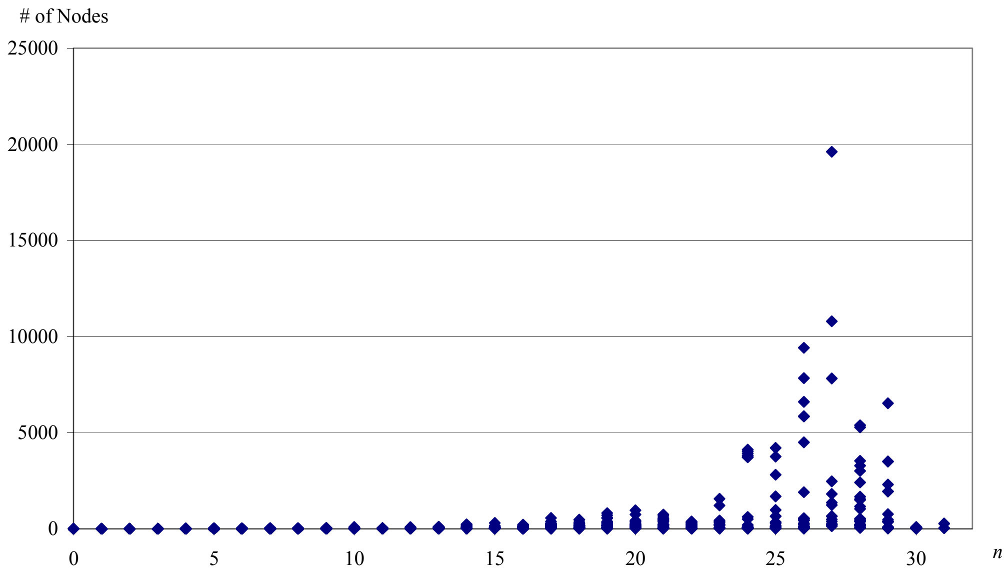


Figure C.2 Number of Nodes versus Number of Tasks

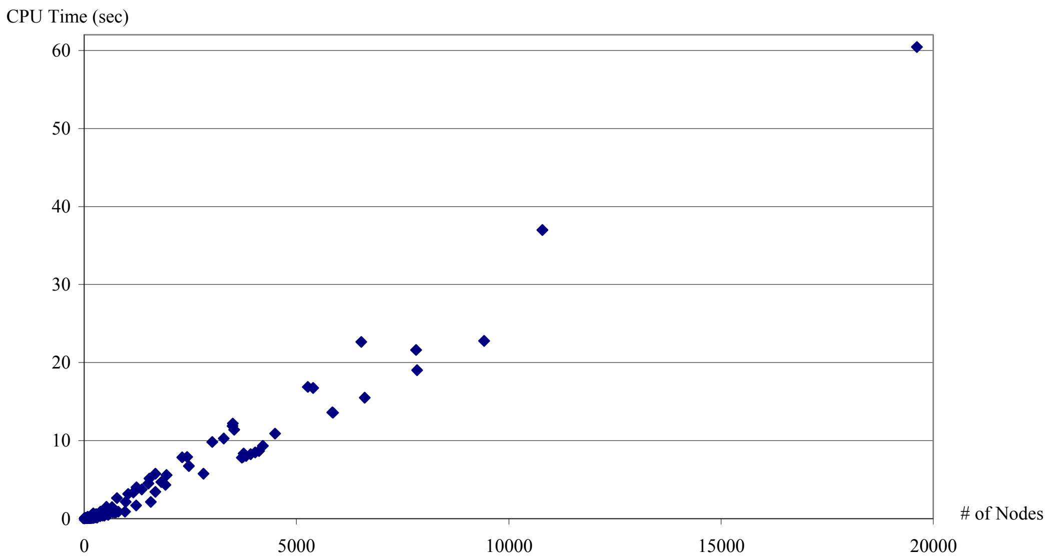


Figure C.3 CPU Time versus Number of Nodes

Table C-9 Results of Large Problem Set With Fixed Number of Nodes

				UB2				UB_BEST			
Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	# of Nodes	Optimal Node	Average CPU Time	Max	# of Nodes	Optimal Node	Average CPU Time	Max
GUNTHER-1-0	35	18	1	13	13	0.0235	0.0320	13	13	0.0000	0.0000
GUNTHER-1-0	35	18	2	299	27	0.0983	0.1090	243	27	0.0969	0.1100
GUNTHER-1-0	35	18	3	611	27	0.3577	0.3600	547	27	0.3859	0.3910
GUNTHER-2-0	35	27	1	19	13	0.0127	0.0160	15	13	0.0047	0.0160
GUNTHER-2-0	35	27	2	1055	25	0.6294	0.6410	1029	25	0.7825	0.7970
GUNTHER-2-0	35	27	3	2881	39	3.4675	3.4880	2823	39	4.2830	4.3320
GUNTHER-3-0	35	35	1	75	21	0.0218	0.0320	45	21	0.0110	0.0160
GUNTHER-3-0	35	35	2	6901	35	10.2831	10.3040	5531	35	10.1392	10.1800
GUNTHER-3-0	35	35	3	10001	61	25.3099	25.3360	10001	61	29.1244	29.1630
KILBRID-1-0	45	23	1	67	27	0.0453	0.0630	67	27	0.0314	0.0320
KILBRID-1-0	45	23	2	67	27	0.0455	0.0470	67	27	0.0578	0.0630
KILBRID-1-0	45	23	3	67	27	0.0921	0.1090	67	27	0.1033	0.1250
KILBRID-2-0	45	34	1	203	35	0.1264	0.1410	203	35	0.1640	0.1720
KILBRID-2-0	45	34	2	1449	47	2.4837	2.5140	1447	45	3.0508	3.0800
KILBRID-2-0	45	34	3	10001	35	35.6650	35.7160	10001	35	40.0337	40.0890
KILBRID-3-0	45	45	1	55	39	0.0608	0.0630	55	39	0.0658	0.0780
KILBRID-3-0	45	45	2	8593	89	30.5426	30.5750	8593	89	35.0726	35.1220
KILBRID-3-0	45	45	3	10001	2337	58.4076	61.1030	10001	2269	64.4042	67.2620
HAHN-1-0	53	27	1	11	11	0.0816	0.0980	11	11	0.0597	0.0630
HAHN-1-0	53	27	2	237	39	0.2212	0.2340	85	39	0.1412	0.1470
HAHN-1-0	53	27	3	723	39	0.8618	0.8860	669	39	1.0048	1.0500
HAHN-2-0	53	40	1	5	0	0.0846	0.0940	5	0	0.0785	0.0820
HAHN-2-0	53	40	2	531	55	0.8557	0.9020	391	55	0.8211	0.8530
HAHN-2-0	53	40	3	7473	55	18.2714	19.0140	7457	55	21.4880	22.3930
HAHN-3-0	53	53	1	31	21	0.1017	0.1110	25	21	0.0940	0.1100

Table C-9 Results of Large Problem Set With Fixed Number of Nodes (Continued)

Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	UB2				UB_BEST			
				# of Nodes	Optimal Node	Average CPU Time	Max	# of Nodes	Optimal Node	Average CPU Time	Max
HAHN-3-0	53	53	2	10001	75	30.3697	32.4530	3427	75	13.0576	15.9660
HAHN-3-0	53	53	3	10001	75	46.3499	47.1210	10001	75	51.7985	52.6780
WARNECKE-1-0	58	29	1	25	11	0.0360	0.0480	25	11	0.0093	0.0160
WARNECKE-1-0	58	29	2	435	11	0.4334	0.4390	343	11	0.4527	0.4620
WARNECKE-1-0	58	29	3	995	11	1.3843	1.4330	867	11	1.5438	1.5940
WARNECKE-2-0	58	44	1	151	35	0.0767	0.0790	135	35	0.0832	0.0940
WARNECKE-2-0	58	44	2	7113	57	14.5575	14.7890	6941	57	17.0335	17.2440
WARNECKE-2-0	58	44	3	10001	59	33.8808	36.8330	10001	59	38.0293	38.3710
WARNECKE-3-0	58	58	1	601	43	0.3635	0.3780	591	43	0.4618	0.4710
WARNECKE-3-0	58	58	2	10001	77	49.1345	49.5790	10001	77	54.4907	54.9940
WARNECKE-3-0	58	58	3	10001	81	81.6108	82.0600	10001	81	88.4167	91.5550
TONGE70-1-0	70	35	1	47	25	0.0671	0.0790	47	25	0.0422	0.0470
TONGE70-1-0	70	35	2	1335	55	1.2411	1.2660	1311	49	1.5349	1.5740
TONGE70-1-0	70	35	3	10001	713	28.5583	28.6770	10001	625	32.7822	32.8830
TONGE70-2-0	70	53	1	81	41	0.1078	0.1100	75	41	0.1125	0.1250
TONGE70-2-0	70	53	2	10001	41	21.9420	22.0010	10001	41	25.9185	25.9780
TONGE70-2-0	70	53	3	10001	41	73.1284	73.3140	10001	41	79.3215	79.5210
TONGE70-3-0	70	70	1	581	63	1.0045	1.0160	573	59	1.2307	1.2360
TONGE70-3-0	70	70	2	10001	561	46.8913	50.0700	10001	561	52.7817	52.8680
TONGE70-3-0	70	70	3	10001	43	180.5759	180.8310	10001	43	190.5049	194.5510
WEE-MAG-1-0	75	38	1	91	59	0.3437	0.3600	91	59	0.3560	0.3600
WEE-MAG-1-0	75	38	2	241	73	0.9763	0.9850	241	73	1.1327	1.1420
WEE-MAG-1-0	75	38	3	241	73	0.9918	1.0020	241	73	1.1546	1.1720
WEE-MAG-2-0	75	57	1	10001	81	63.5440	67.4790	10001	81	69.8994	70.2960
WEE-MAG-2-0	75	57	2	10001	383	96.8630	97.4900	10001	381	104.2346	104.8910
WEE-MAG-2-0	75	57	3	10001	377	117.4800	117.8770	10001	377	125.8997	126.2150
WEE-MAG-3-0	75	75	1	10001	3275	113.8785	114.4400	10001	3275	120.9858	122.8860

Table C-9 Results of Large Problem Set With Fixed Number of Nodes (Continued)

Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	UB2				UB_BEST			
				# of Nodes	Optimal Node	Average CPU Time	Max	# of Nodes	Optimal Node	Average CPU Time	Max
WEE-MAG-3-0	75	75	2	10001	5741	159.8308	160.1520	10001	5717	166.9123	169.6860
WEE-MAG-3-0	75	75	3	10001	17	291.5248	292.9560	10001	17	302.0146	306.8170
ARC83-1-0	83	42	1	181	41	0.2095	0.2190	155	41	0.2049	0.2180
ARC83-1-0	83	42	2	1225	63	3.1321	3.1700	1079	63	3.3276	3.3700
ARC83-1-0	83	42	3	10001	65	48.8810	49.1640	10001	65	54.8728	57.6690
ARC83-2-0	83	63	1	1121	75	2.0297	2.0370	1097	75	2.5306	2.5650
ARC83-2-0	83	63	2	10001	101	67.1244	67.3320	10001	101	74.0077	74.2250
ARC83-2-0	83	63	3	10001	123	114.0166	114.1600	10001	123	122.2189	123.4190
ARC83-3-0	83	83	1	5759	77	20.3256	20.4570	5585	77	23.1492	23.1740
ARC83-3-0	83	83	2	10001	123	150.6083	150.7660	10001	123	160.1594	163.2800
ARC83-3-0	83	83	3	10001	153	248.7691	252.3990	10001	153	259.2536	263.7920
LUTZ2-1-0	89	45	1	69	45	0.4306	0.4530	67	45	0.4054	0.4060
LUTZ2-1-0	89	45	2	10001	39	31.9072	32.0140	10001	39	36.7276	36.8310
LUTZ2-1-0	89	45	3	10001	39	48.6590	48.7800	10001	39	54.6663	57.5940
LUTZ2-2-0	89	67	1	111	0	0.7461	0.7500	97	0	0.7478	0.7510
LUTZ2-2-0	89	67	2	10001	303	82.1478	82.2950	10001	303	89.0364	89.2100
LUTZ2-2-0	89	67	3	10001	303	99.2918	101.3590	10001	303	106.5122	106.6920
LUTZ2-3-0	89	89	1	385	41	1.1335	1.1420	385	41	1.2226	1.2360
LUTZ2-3-0	89	89	2	10001	63	152.0819	155.0750	10001	63	161.7054	162.4380
LUTZ2-3-0	89	89	3	10001	63	218.1216	247.3570	10001	63	228.6664	252.0360
LUTZ3-1-0	89	45	1	81	35	0.4526	0.4690	69	35	0.4388	0.4490
LUTZ3-1-0	89	45	2	10001	57	33.0656	33.7440	10001	57	37.8675	38.6740
LUTZ3-1-0	89	45	3	10001	91	50.7161	53.7570	10001	91	55.8377	59.5670
LUTZ3-2-0	89	67	1	135	49	0.8857	0.9430	101	49	0.8767	0.9430
LUTZ3-2-0	89	67	2	10001	129	72.2211	77.1500	10001	129	78.0026	81.3560
LUTZ3-2-0	89	67	3	10001	407	112.4901	115.3070	10001	399	120.3045	125.1630
LUTZ3-3-0	89	89	1	1271	65	4.3150	4.3590	875	65	3.7568	3.8380

Table C-9 Results of Large Problem Set With Fixed Number of Nodes (Continued)

Test Problem	Number of Tasks	Number of Parts	Cycle Time Level	UB2				UB_BEST			
				# of Nodes	Optimal Node	Average CPU Time	Max	# of Nodes	Optimal Node	Average CPU Time	Max
LUTZ3-3-0	89	89	2	10001	91	121.3894	122.8800	10001	91	129.4648	133.0790
LUTZ3-3-0	89	89	3	10001	91	227.8950	232.7970	10001	91	239.8805	245.1690
MUKHERJE-1-0	94	47	1	129	51	0.2386	0.2660	89	51	0.2310	0.2350
MUKHERJE-1-0	94	47	2	9935	77	49.1780	49.2970	9915	77	56.0290	57.4650
MUKHERJE-1-0	94	47	3	9935	77	49.4310	51.2770	9915	77	56.1018	57.7100
MUKHERJE-2-0	94	71	1	477	77	0.9147	0.9330	271	77	0.7783	0.8040
MUKHERJE-2-0	94	71	2	10001	273	168.9906	170.5270	10001	249	179.2536	182.3550
MUKHERJE-2-0	94	71	3	10001	273	168.8513	169.0300	10001	249	178.8249	179.0140
MUKHERJE-3-0	94	94	1	1771	93	6.0317	6.0510	1347	93	5.8931	5.9060
MUKHERJE-3-0	94	94	2	10001	1263	306.6071	309.2580	10001	1263	319.8166	322.7180
MUKHERJE-3-0	94	94	3	10001	1263	306.9406	309.6820	10001	1263	319.4852	322.0360

Table C-10 B&B results for Problem I make span analysis (Continued)

	PROBLEM I				PROBLEM II						
Test Problem	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	Average Reduction in Number of Nodes	Average Reduction in CPU Times	Average Deviation of Optimal Solution Values
BOWMAN8-1-0	8.2	5.2	0.014	17.7	4.0	4.0	0.006	13.5	51.2	55.3	27.2
BOWMAN8-2-0	15.0	6.2	0.011	12.3	5.2	4.8	0.006	9	65.3	42.2	36.6
BOWMAN8-3-0	17.4	11.6	0.011	11.9	9.8	8.4	0.008	10.2	43.7	29.1	12.5
BUXEY-1-0	85.2	19.0	0.052	44.4	24.0	15.8	0.016	37.8	71.8	69.8	22.0
BUXEY-2-0	381.8	38.6	0.342	32.9	123.2	27.8	0.147	29.9	67.7	57.1	11.3
BUXEY-3-0	2982.6	223.2	5.072	23.1	331.4	29.9	0.605	13.5	88.9	88.1	43.3
MERTENS-1-0	7.0	4.3	0.011	3.9	3.0	2.7	0.008	2.6	57.1	28.4	54.0
MERTENS-2-0	14.8	4.0	0.011	1.8	5.0	2.6	0.008	1.1	66.2	28.4	31.7
MERTENS-3-0	20.4	7.2	0.011	2.4	7.4	4.0	0.008	1.2	63.7	28.4	40.8
HESKIA-1-0	19.6	18.0	0.020	407.9	17.4	17.4	0.011	400.6	11.2	46.1	1.8
HESKIA-2-0	23.6	23.6	0.023	356.5	24.0	23.8	0.016	356.5	-1.7	33.3	0.0
HESKIA-3-0	222.2	41.8	0.261	310.1	29.2	29.2	0.023	310.1	86.9	91.0	0.0
JACKSON-1-0	9.4	7.7	0.012	11	6.6	6.6	0.008	8.3	29.8	37.6	21.9
JACKSON-2-0	16.6	11.2	0.012	7.7	10.4	9.6	0.009	6.2	37.3	24.8	29.6
JACKSON-3-0	20.8	14.8	0.014	6.5	15.4	14.0	0.011	6.3	26.0	22.7	8.3
JAESCHKE-1-0	9.0	6.9	0.011	7.8	5.0	4.2	0.005	6	44.4	56.9	34.4
JAESCHKE-2-0	13.8	7.8	0.012	4.9	6.0	5.4	0.008	3.4	56.5	37.6	39.3
JAESCHKE-3-0	23.4	8.4	0.011	3.7	4.8	3.3	0.008	1.1	79.5	28.4	78.3
LUTZ1-1-0	194.0	16.2	0.095	1834.4	13.4	10.2	0.012	1374.7	93.1	86.9	29.7
LUTZ1-2-0	1551.4	31.0	1.634	1719.2	121.8	21.2	0.089	1020.9	92.1	94.6	39.6
LUTZ1-3-0	20184.6	34.0	38.315	1230.6	262.2	13.4	0.397	821.1	98.7	99.0	32.8
MANSOOR-1-0	7.4	7.2	0.011	32.6	3.8	3.3	0.005	8.3	48.6	56.4	75.7
MANSOOR-2-0	12.4	12.2	0.011	49.3	9.2	9.0	0.008	26.7	25.8	28.4	45.2
MANSOOR-3-0	17.0	17.0	0.014	51.8	14.4	14.2	0.009	39.1	15.3	33.3	24.3
MITCHELL-1-0	67.4	13.8	0.028	25.5	12.2	10.4	0.009	17.9	81.9	66.9	30.1

Table C-10 B&B results for Problem I make span analysis (Continued)

	PROBLEM I				PROBLEM II						
Test Problem	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	Average Reduction in Number of Nodes	Average Reduction in CPU Times	Average Deviation of Optimal Solution Values
MITCHELL-2-0	280.4	26.6	0.142	19.3	22.4	15.0	0.014	16.2	92.0	90.1	20.3
MITCHELL-3-0	1129.2	55.8	0.907	15	39.0	18.6	0.022	14.1	96.5	97.6	5.7
ROSZIEG-1-0	122.6	16.0	0.053	27.8	11.4	10.6	0.009	22.5	90.7	82.5	19.1
ROSZIEG-2-0	834.2	18.0	0.577	15.6	40.6	12.4	0.030	13.5	95.1	94.9	13.1
ROSZIEG-3-0	6286.0	24.0	7.520	14	91.4	15.2	0.064	12.4	98.5	99.1	10.9
SAWYER30-1-0	51.2	19.8	0.031	67.6	23.0	17.0	0.016	62.9	55.1	50.2	7.3
SAWYER30-2-0	246.2	37.0	0.233	35	58.6	21.4	0.058	33.4	76.2	75.1	5.6
SAWYER30-3-0	2219.2	168.8	3.760	29.1	338.8	36.0	0.627	28.8	84.7	83.3	1.0
AKKSKT12-1-0	7.6	7.0	0.012	69.5	7.0	7.0	0.008	69.5	7.9	37.6	0.0
AKKSKT12-2-0	11.8	9.0	0.011	69.3	9.0	9.0	0.008	69.3	23.7	28.4	0.0
AKKSKT12-3-0	15.4	10.8	0.012	64.7	11.0	9.8	0.008	63.9	28.6	36.8	1.1
AKO20T-1-0	12.6	6.6	0.012	20.1	3.6	3.4	0.006	12.2	71.4	50.4	42.6
AKO20T-2-0	42.2	20.6	0.017	18	13.8	9.6	0.009	12.2	67.3	45.7	41.0
AKO20T-3-0	113.8	39.6	0.039	13.5	41.8	16.5	0.023	10.8	63.3	40.2	24.9
AKO30T1-1-0	24.6	14.0	0.014	39.4	11.2	11.2	0.008	39.4	54.5	44.3	0.0
AKO30T1-2-0	49.4	25.0	0.024	29.7	20.0	17.6	0.012	29.7	59.5	46.8	0.0
AKO30T1-3-0	118.0	36.4	0.058	21.2	32.6	22.4	0.017	21.2	72.4	70.2	0.0
AKO30T2-1-0	14.0	7.9	0.014	27.8	7.4	7.0	0.008	22.2	47.1	44.7	17.7
AKO30T2-2-0	27.4	20.6	0.016	19.9	20.2	16.4	0.009	19.4	26.3	40.1	2.5
AKO30T2-3-0	49.4	31.6	0.023	19	16.4	13.6	0.011	18.4	66.8	53.4	2.6
GUN17T-1-0	19.2	11.6	0.014	66.4	13.2	11.6	0.009	66.4	31.3	33.3	0.0
GUN17T-2-0	46.2	19.8	0.019	76.6	21.4	14.8	0.011	76.6	53.7	41.5	0.0
GUN17T-3-0	66.4	23.6	0.036	64.4	24.4	17.4	0.014	64.4	63.3	61.1	0.0
GUN8T-1-0	8.2	3.7	0.012	13.1	2.8	2.8	0.008	9.6	65.9	37.6	25.3
GUN8T-2-0	14.4	6.4	0.011	13.6	3.2	2.7	0.008	7	77.8	28.4	61.5

Table C-10 B&B results for Problem I make span analysis

	PROBLEM I				PROBLEM II						
Test Problem	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	# of Nodes	Optimal Node	Average CPU Time	Optimal Solution Value	Average Reduction in Number of Nodes	Average Reduction in CPU Times	Average Deviation of Optimal Solution Values
GUN8T-3-0	38.8	13.4	0.012	10.5	3.8	3.8	0.006	6.1	90.2	49.6	44.4
LAM20T-1-0	8.0	6.7	0.012	57.1	7.2	6.8	0.008	57.1	10.0	37.6	0.0
LAM20T-2-0	12.0	10.0	0.009	66	8.8	7.8	0.008	59.3	26.7	17.0	10.1
LAM20T-3-0	7.2	6.0	0.012	65.5	4.2	4.2	0.006	60.7	41.7	49.6	7.1
LAM30T-1-0	8.2	6.8	0.014	7.2	5.4	3.4	0.008	3.5	34.1	44.7	59.6
LAM30T-2-0	10.6	8.0	0.011	10.4	4.4	4.4	0.006	5.2	58.5	43.1	51.1
LAM30T-3-0	16.4	10.6	0.012	7.9	5.8	5.2	0.008	4.5	64.6	37.6	42.8
MAS30T-1-0	95.2	36.8	0.053	388.6	29.6	18.6	0.020	388.6	68.9	61.7	0.0
MAS30T-2-0	578.4	61.2	0.530	249.7	68.0	23.6	0.047	249.1	88.2	91.2	0.1
MAS30T-3-0	8021.0	224.0	11.654	194.5	101.0	29.4	0.089	194.5	98.7	99.2	0.0
MGG7T-1-0	4.8	4.2	0.009	5.9	3.2	3.2	0.006	4.7	33.3	34.0	25.2
MGG7T-2-0	7.4	4.9	0.009	4.8	3.4	3.4	0.008	3.4	54.1	15.1	24.3
MGG7T-3-0	10.0	6.0	0.011	5.8	5.6	4.4	0.006	4.6	44.0	43.6	15.5
WANG18T-1-0	15.4	10.6	0.014	23.1	10.4	9.2	0.008	21.1	32.5	44.7	9.6
WANG18T-2-0	38.8	14.0	0.017	27.7	16.4	14.0	0.011	27.3	57.7	35.7	2.2
WANG18T-3-0	136.2	17.6	0.077	29.1	34.0	17.6	0.027	29.1	75.0	65.4	0.0
YKA19T-1-0	13.0	8.4	0.016	7.3	8.8	7.8	0.008	6.7	32.3	50.3	9.8
YKA19T-2-0	23.6	13.2	0.016	11.3	15.0	12.2	0.011	10.3	36.4	29.5	10.3
YKA19T-3-0	49.8	12.8	0.028	9.2	23.4	13.2	0.020	8.2	53.0	27.8	9.4
YKA27T-1-0	19.8	14.2	0.019	17.9	18.6	14.2	0.014	17.9	6.1	25.1	0.0
YKA27T-2-0	27.0	18.4	0.024	17.7	25.4	17.6	0.025	17.1	5.9	-6.4	5.5
YKA27T-3-0	32.2	25.6	0.031	21.1	31.4	25.4	0.036	20.8	2.5	-15.0	1.6
YKA31T-1-0	62.6	16.2	0.042	12.7	36.8	15.6	0.039	11.7	41.2	7.6	6.3
YKA31T-2-0	230.0	25.0	0.208	11.5	117.4	24.6	0.180	11.1	49.0	13.6	3.4
YKA31T-3-0	397.8	27.2	0.536	11.9	109.6	25.2	0.205	11.3	72.4	61.8	5.2