MULTI-OBJECTIVE ASSEMBLY JOB SHOP SCHEDULING USING GENETIC ALGORITHM AND TABU SEARCH

A THESIS

Submitted by

DILEEPLAL J.

for the award of the degree of

DOCTOR OF PHILOSOPHY

Under the Faculty of Technology

DEPARTMENT OF SHIP TECHNOLOGY

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

KOCHI – 682 022

AUGUST 2012

Dr. K. P. Narayanan
Associate Professor and Head,
Department of Ship Technology,
Cochin University of Science and Technology,
Kochi – 682 022.

August 3, 2012

CERTIFICATE

This is to certify that the thesis entitled **"Multi-Objective Assembly Job Shop Scheduling using Genetic Algorithm and Tabu Search"** submitted by **Shri. Dileeplal J.** to Cochin University of Science and Technology in partial fulfillment of the requirement for the award of the degree of Doctor of Philosophy in the Faculty of Technology is a bonafide record of the research work carried out by him under my supervision. The contents of this thesis have not been submitted to any other university or institute for the award of any degree.

Dr. K. P. Narayanan
(Supervising Guide)

DECLARATION

I do declare that the thesis titled **"Multi-Objective Assembly Job Shop Scheduling using Genetic Algorithm and Tabu Search"** submitted to Cochin University of Science and Technology in partial fulfillment of the requirement for the award of the degree of Doctor of Philosophy in the Faculty of Technology is a bonafide record of the research work carried out by me. The contents of this thesis have not been submitted to any other university or institute for the award of any degree.

Kochi,
August 3, 2012.

Dileeplal J.
Researh Scholar (Part Time),
Reg. No. 2948,
Department of Ship Technology,
CUSAT, Kochi – 682 022.

ACKNOWLEDGMENTS

At the onset let me praise and thank GOD THE ALMIGHTY for showering me Thy choicest blessings without which this work would have been impossible.

I wish to express my deep sense of respect and gratitude to my supervising guide Dr. K. P. Narayanan, Head, Department of Ship Technology, CUSAT, who has been a constant source of inspiration during the course of my research work. I am indebted to him for his valuable guidance and constant support.

I whole heartedly express my sincere gratitude to Dr. C. B. Sudheer, Department of Ship Technology, CUSAT, for the valuable guidance, suggestions and help he has extended to me during the period of research. He has always been patient towards my shortcomings and kept encouraging me to work in a better way.

I express my deep-felt gratitude to my friend Dr. B. S. Girish, Assistant Professor, Department of Aerospace Engineering, Indian Institute of Space Science and Technology, Thiruvananthapuram. I owe him for instilling the spirit of research in me and always bringing my confidence up with his words of appreciation.

I place on record my profound gratitude to Dr. C. G. Nandakumar and all faculty members of Department of Ship Technology, CUSAT for their valuable comments and encouragement. I offer my regards to all non-teaching staff of the Department of Ship Technology, CUSAT for their kind cooperation.

I am grateful to the Secretary, Mar Athanasius College Association and Principal, Mar Athanasius College of Engineering, Kothamangalam for their support. I am also obliged to all my colleagues at Mar Athanasius College of Engineering, Kothamangalam for the love and care they had shown to me.

With endless respect and love, I thank all my teachers who gave me light in life through education.

With immense pleasure, I convey my gratitude to my friends for their love, help, encouragement and motivation.

The encouragement, prayers and blessings of my family members are deeply acknowledged. My unbound gratitude goes to my beloved mother who took a lot of pain to help me and blessed me with her prayers.

I am deeply indebted to my wife Smitha and my daughter Diya for their support, understanding, tolerance, and unlimited patience without which I would not have completed this study.

DILEEPLAL J.

ABSTRACT

Assembly job shop scheduling problem (AJSP) is one of the most complicated combinatorial optimization problem that involves simultaneously scheduling the processing and assembly operations of complex structured products. The problem becomes even more complicated if a combination of two or more optimization criteria is considered. This thesis addresses an assembly job shop scheduling problem with multiple objectives. The objectives considered are to simultaneously minimizing makespan and total tardiness.

In this thesis, two approaches viz., weighted approach and Pareto approach are used for solving the problem. However, it is quite difficult to achieve an optimal solution to this problem with traditional optimization approaches owing to the high computational complexity. Two metaheuristic techniques namely, genetic algorithm and tabu search are investigated in this thesis for solving the multi-objective assembly job shop scheduling problems.

Three algorithms based on the two metaheuristic techniques for weighted approach and Pareto approach are proposed for the multi-objective assembly job shop scheduling problem (MOAJSP). A new pairing mechanism is developed for crossover operation in genetic algorithm which leads to improved solutions and faster convergence.

The performances of the proposed algorithms are evaluated through a set of test problems and the results are reported. The results reveal that the proposed algorithms based on weighted approach are feasible and effective for solving MOAJSP instances according to the weight assigned to each objective criterion and the proposed algorithms based on Pareto approach are capable of producing a number of good Pareto optimal scheduling plans for MOAJSP instances.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS AND NOTATIONS

a	Scaling constant for crossover pairing mechanism
AJS	Assembly job shop scheduling
AJSP	Assembly job shop scheduling problem
ANOVA	Analysis of variance
AS_p	Set of assembly operations of product p
BOM	Bill of materials
C	Component
C_{pij}	Completion time of operation O_{pij}
CS_p	Set of components of product p
CUDA	Compute unified device architecture
D_p	Due date of product p
E_{pij}	Set of preceding operations O_{pij} of assembly operation j of product p
F_p	Finish time of product p
FIFO	First in first out
GA	Genetic algorithm
H	A large positive integer
i, i'	Index for components ($i=1, 2, \ldots\ldots, n_p$)
j, j'	Index for operations ($j=1, 2, \ldots\ldots, J_{pi}$)
J_{pi}	Number of processing operations of component i of product p
JDD	Job due date
l	Length of tabu list

L_m	Set of processing operations O_{pij} that can be assigned to machine m
L_p	Lateness of product p
m	Index for machines ($m=1, 2, \ldots\ldots, M$)
M	Number of machines
MOAJS	Multi-objective assembly job shop scheduling
MOAJSP	Multi-objective assembly job shop scheduling problem
MOGA	Multi-objective genetic algorithm
MOHA	Multi-objective hybrid genetic algorithm
MOJS	Multi-objective job shop scheduling
MOTS	Multi-objective tabu search
N	Number of products
N_{it}	Total number of iterations for an algorithm
NSGA II	Non dominated sorting genetic algorithm II
n_p	Number of components of product p
O_{pij}	Operation j of component i of product p
ODD	Operation due date
P	Product
p, p'	Index for products ($p=1, 2, \ldots\ldots, N$)
PAGA	Pareto archived genetic algorithm
PAHA	Pareto archived hybrid genetic algorithm
PATS	Pareto archived tabu search
P_c	Probability of crossover
PDR	Priority dispatching rules

P_m	Probability of mutation
POX	Precedence preserving order based crossover
PRS	Percentage reduction in solution
PT	Product type
PV	Pairing value of chromosomes
S_{pij}	Start time of operation O_{pij}
SPT	Shortest processing time
T_p	Tardiness of product p
TS	Tabu search
TWKR	Total work content remaining
t_{pijm}	Processing/assembly time of operation O_{pij} on machine m
w_1	Weight assigned to makespan criterion
w_2	Weight assigned to total tardiness criterion
WIP	Work in process
$X_{pijp'i'j'm}$	Decision variable for generating a sequence between operations O_{pij} and $O_{p'i'j'}$ for loading on machine m

CHAPTER 1
OUTLINE OF RESEARCH

1.1 INTRODUCTION

Scheduling can be defined as the allocation of resources over time to perform a collection of tasks (Baker, 1974). It has a wide area of application in all economic domains in the real world scenario. Scheduling problems usually arise in various business environments such as production, transportation, distribution and information processing environments.

In real manufacturing environment, there is a requirement to use the available resources as efficiently as possible. An effective schedule enables the organization to utilize its resources effectively and helps to attain its strategic objectives. Scheduling in the context of manufacturing systems refers to the determination of the sequence in which jobs are to be processed over the production stages, followed by the determination of the start time and finish time of processing jobs (Conway et al., 1967). The importance of scheduling has increased in recent years due to the growing consumer demand for variety, reduced product life cycles, changing markets with global competition and rapid development of new processes and technologies (Ho et al., 2007). Besides this, the generation of consistently good schedules has proven to be extremely difficult in medium to large shops and optimal scheduling involves costly and impractical enumeration procedures.

The production scheduling problem encapsulates many variations such as single machine scheduling, parallel machine scheduling, flow shop scheduling and job shop scheduling problems. Each of these problem classes is unique, and each has its own constraints and objectives. Job shop scheduling is the most common and difficult combinatorial optimization problem among them. It is concerned with determining the release order and times of a set of jobs on the relevant machines subject to the processing constraints, in an effort to improve the production efficiency and reduce the processing duration so as to gain as high profits as possible (Zhou et al., 2001). The analysis of job shop scheduling problems provides important insights into the solution of scheduling problems encountered in more realistic and complicated systems (Pinedo, 2005). Besides this, most of the real world manufacturing companies continue to experience difficulties with their specific job shop scheduling problems. Accordingly, job shop scheduling problems still receive ample attention from both researchers and practitioners. In this viewpoint, this thesis focuses on scheduling in manufacturing systems, particularly job shops.

1.2 ASSEMBLY JOB SHOP SCHEDULING

An assembly job shop is an extension of conventional job shop. It is frequently employed for the manufacture of low volume high variety products involving both processing and assembly operations. The assembly job shop comprises a machine shop and an assembly shop. The machine shop often consists of a set of machines organized based on their functionality into various work centers and an assembly shop often consists of a set of assembling stations. The final product in an assembly job shop has a set of components and subassemblies that assemble together to build up the end product. In the assembly job shop, components and/or subassembly undergo operations in a particular order as per the precedence constrains in the machine shop and wait for the arrival of its mating components at the assembly shop, for the assembly operations to start. Therefore, the operations in an assembly job shop include both serial and parallel operations. In addition to the waiting time for a resource (machine/operator), scheduling in an assembly shop requires considering the time an item may have to wait for its parallel component before the required assembly operations can take place. Hence, the lead time of a product consists of combination of the following: the flow time of its components, the assembly time, the waiting time of assembly operations at assembly stations, and the staging delay at various assembly points in the product (Philipoom et al., 1989). Scheduling an assembly job shop requires proper coordination of materials flow through the various stages necessary to complete a product (Kolisch, 2001). These additional scheduling considerations underline the complexity of scheduling in assembly job shops when compared to the conventional job shops. In this context, this research is directed towards the scheduling of assembly job shops which is an important task for manufacturing industry in terms of improving machine utilization and meeting due dates.

1.3 MOTIVATION

The assembly job shop scheduling is quite challenging and frequently encountered in the industrial environment. Furthermore, instead of considering only a single objective, such scheduling problems in practice involve simultaneous optimization of several competing objectives that induces additional complexity in solving the problem to optimality. Despite their importance, scant attention has been given to multiple objective assembly job shop scheduling and the previous research seldom considered a multiple criterion to optimize. It is quite difficult to attain an optimal solution to this scheduling problem with traditional optimization approaches owing to the high computational complexity. Therefore, heuristic approaches are commonly preferred over the traditional mathematical techniques for solving such complex combinatorial optimization problems. In light of the above, this research aims to develop efficient heuristics to solve multi-objective assembly job shop scheduling problem.

1.4 OBJECTIVES OF THE THESIS

In order to achieve the aim, the following objectives have been set:

- Modeling multi-objective assembly job shop scheduling problem.
- Defining the problems for the proposed model.
- To identify and design suitable heuristics to be applied.
- Comparison and evaluation of the proposed heuristics based on objectives to obtain optimal or near optimal schedules.

1.5 MULTI-OBJECTIVE OPTIMIZATION

Most real world optimization problems naturally involve multiple and conflicting objectives. Hence, an optimum solution with respect to one objective often results in undesirable solution with respect to other objectives. Therefore, a reasonable answer to a multi-objective problem is to explore a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution. However, in practical standpoint there is only one solution required for an optimization problem, no matter whether it involves multiple objectives or not. Generally, there are preference-based multi-objective optimization procedures and ideal multi-objective optimization procedures to handle these optimization problems that require multi-objective analysis (Deb, 2001).

The preference-based multi-objective optimization procedure is based on weighted approach. It begins with choosing a preference vector based on the higher level information, in which the weights for different objective criterions are included. Subsequently, this vector is used to construct a composite function, which is then optimized to find a single trade-off optimal solution. This approach also can be used to find multiple trade-off solutions with different preference vectors.

The ideal multi-objective optimization procedure is based on Pareto approach, which is used to find out a set of trade-off optimal solutions. The multiple trade-off solutions are first found out and higher level information is then used to choose one of the trade-off solutions.

1.6 HEURISTIC SOLUTION TECHNIQUES

A large number of heuristic solution techniques for scheduling problems have been reported in the literature, with varying degrees of success. These include dispatching rules, artificial intelligence techniques, local search techniques and metaheuristic techniques such as neighbourhood based heuristics and population based heuristics. The classification of heuristic solution techniques for scheduling problems are given in Figure 1.1. The salient remarks concerning these heuristic techniques are given below (Hutchison, 1991, Brucker, 1995, Consoli, 2006, Girish, 2009):

- Dispatching rules producing results in short time, but they are questionable for stability and environment dependence.
- Artificial intelligence techniques such as expert systems and neural networks use both quantitative and qualitative knowledge in the decision making process. They generate only feasible solutions and can be time consuming to build and verify, as well as difficult to maintain and change.
- Local search techniques are known for producing excellent results in short run times, but they are susceptible of getting stuck in local optima.
- The neighbourhood based heuristics such as tabu search (TS) and simulated annealing (SA) extend local search to escape from local optima and approach new areas of attraction in the search space to reach optimum/near optimum solution. The performance of TS and SA are dependent on initial solutions and good initialization lead to excellent final solution in short computing times.

- The population based heuristics such as genetic algorithm (GA), ant colony algorithm (ACO) and particle swarm optimization (PSO), which belongs to the random search strategy, guarantees near optimal solutions in actual cases.

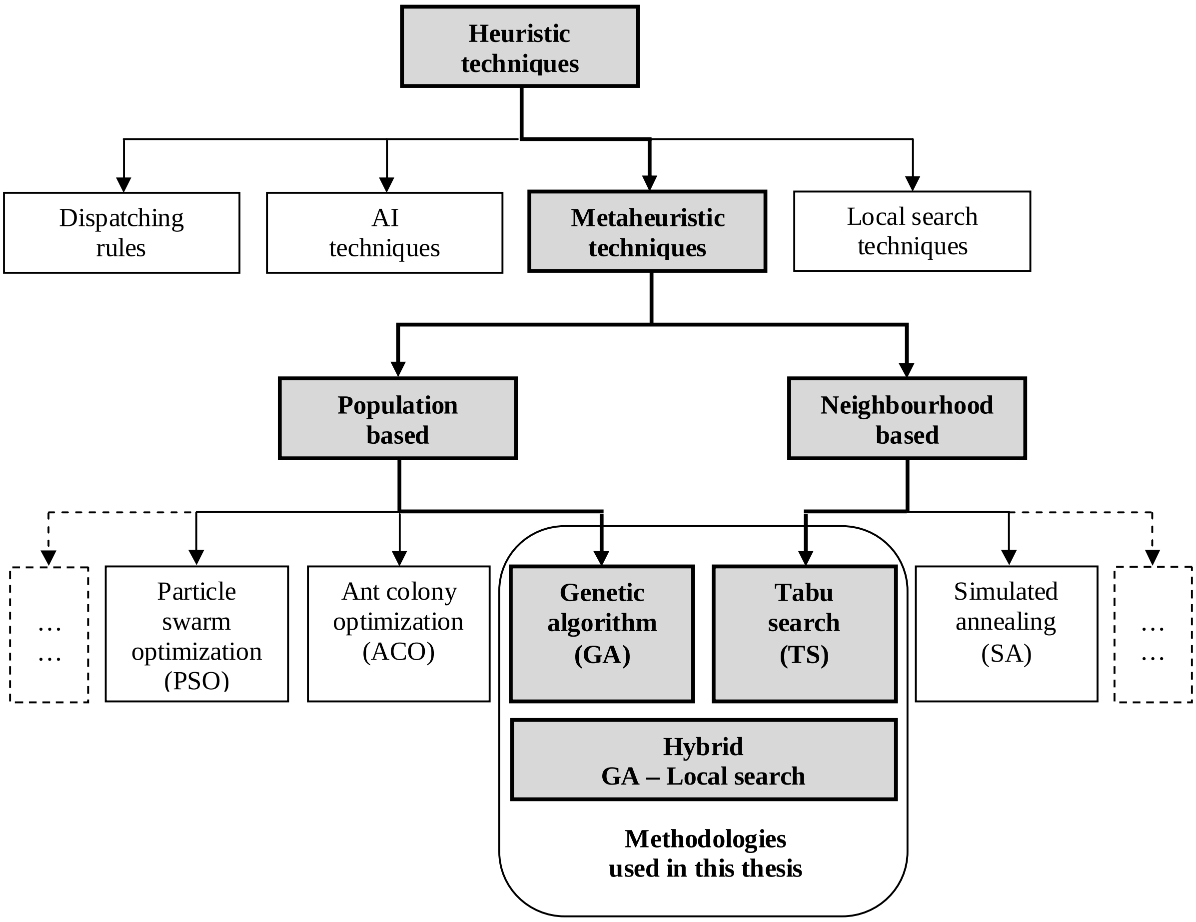

Figure 1.1 Classification of heuristic solution techniques

Among the above heuristic techniques, metaheuristics have demonstrated their ability to solve any complex combinatorial optimization problems. Therefore, metaheuristic techniques such as genetic algorithm (GA) and tabu search (TS) are considered in this thesis for solving multi-objective assembly job shop scheduling problems.

1.7 CONTRIBUTIONS OF THE THESIS

In this thesis different heuristic based scheduling methods have been proposed for assembly job shop scheduling problems with multiple performance measures. The main contributions of this thesis are summarized as follows:

- A description and formulation of multi-objective assembly job shop scheduling problems for simultaneously minimizing makespan and total tardiness. It is the one of the most difficult scheduling problem amongst manufacturing systems. In this thesis, two approaches viz., weighted approach and Pareto approach are used for solving the problem.
- Development of three algorithms based on weighted approach viz., multi-objective genetic algorithm (MOGA), multi-objective tabu search (MOTS) and multi-objective hybrid genetic algorithm (MOHA) for multi-objective assembly job shop scheduling problems.
- Development of three algorithms based on Pareto approach viz., Pareto archived genetic algorithm (PAGA), Pareto archived tabu search (PATS) and Pareto archived hybrid genetic algorithm (PAHA) for multi-objective assembly job shop scheduling problems.
- Development of a pairing mechanism for crossover operation in MOGA, MOHA & PAHA, which leads to improved solutions and faster convergence.
- An experimental study has been conducted for a set of problem instances with complex configurations to check the feasibility and performance of the different algorithms. The problem size is varied from 3 products × 10 machines (62 operations) to 10 products × 15 machines (268 operations).

1.8 ORGANIZATION OF THE THESIS

This chapter discussed the general outline of the research which includes a brief introduction, a short note on assembly job shop scheduling, motivation of the study, objectives of the thesis, multi-objective optimization, heuristic solution techniques and contributions of the thesis. The rest of the thesis is organized as follows:

- Chapter 2 presents the literature review.
- Chapter 3 addresses the problem description.
- Chapter 4 describes the proposed algorithms with weighted approach.
- Chapter 5 describes the proposed algorithms with Pareto approach.
- Chapter 6 presents results and discussions.
- Chapter 7 gives concluding remarks with scope for future research.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The theory of scheduling has received a great deal of attention from academic researchers and practitioners since the early 1950s. The problem of scheduling in assembly job shops is an important operational problem in view of its complexity and practical significance. Additional complexity arises when combining several optimization criteria for evaluating schedule goodness. While a number of research studies have investigated the problem of scheduling in flow shops and job shops, only a few attempts have been done to study the problem of scheduling in assembly job shops that manufacture multilevel jobs. Considering the complexity in the scheduling process of assembly job shops, the theoretical research in this area has become more significant.

This chapter presents a comprehensive review of the literature dealing with various studies carried out on assembly job shop scheduling and multi-objective job shop scheduling.

2.2 ASSEMBLY JOB SHOP SCHEDULING

Generally, there are two types of assembly job shop scheduling problems (Wong et al, 2009). The first type allows machining to the root components only and not the assemblies and the second type allows machining both to the root components and the assemblies i.e., assemblies are also required to be processed on machines before final product assembly. In this section, the various assembly job shop scheduling approaches in the literature are reported.

Sculli (1980) presented the results of an experimental investigation on the priority dispatching rules for a job shop with assembly operations. The study was directed towards rules that utilize job status information such as operation float, number of parts completed, and number of operations remaining on each part which attempt to co-ordinate the completion time of parts required in the same job. Their results indicate that job status information improves most of the measures of performance used.

Sculli (1987) reported the results of a job shop computer simulation study in which jobs were made up of several parts, and where operations on the different parts can be performed on simultaneously on different machining centres. The time required for the assembly operations is taken as negligible. The various performance measures used for evaluation are mean and standard deviation of job flow time, mean and standard deviation of coordination time, mean and standard deviation of job lateness, percentage of jobs late and maximum number of jobs in the system. They made an attempt to classify different priority rules in terms of their operational significance.

Adam et al. (1987) developed a class of priority assignment procedures for job shops processing multi-level assemblies that is aimed at reducing staging delay. They also studied the impact of structural complexity of jobs on the priority assessment rules to reduce job lead time.

Fry et al. (1989) examined the effect of three different product structures on the performance of selected priority dispatching rules in a six machine assembly shop. Ten different BOMs (Bill of materials) are considered to represent product structures that are flat, tall and complex and all the BOMs were processed together in the shop instead of processing individually. An ANOVA was performed to determine the significance of sequencing rules on each of the ten BOMs and each of the four performance measures, mean flow time, mean tardiness, mean absolute lateness and percentage tardy. They showed through a simulation study that there is a significant relationship between product structure and sequencing rule performance.

Philipoom et al. (1989) observed the performance of fourteen due date oriented sequencing rules in a simulated ten machine multistage job shop having one assembly work center. Each due date based sequencing rule is tested using three methods of setting due date milestones such as job due dates, assembly due dates and operation due dates to control the progression of a job toward completion.

Philipoom et al. (1991) proposed multi-attribute based sequencing rules for scheduling assembly job shops. They evaluated the performance of eight sequencing rules on three distinct set of product structures by using four measures of system inventory and four measures of job tardiness.

Doctor et al. (1993) addressed the problem of scheduling a set of jobs with a maximum of three levels of assembly in a make to order job shop environment. They developed a heuristic algorithm to solve the problem for the objective of maximizing machine utilization subject to satisfying job due date requirements.

Adam et al. (1993) considered an assembly job shop where the due dates of arriving jobs were set internally, and jobs had to be scheduled to meet the assigned due dates. They introduced dynamically updated due date assignment procedures where the appropriate coefficients are continually updated to reflect the changing job mix, work load and resources of a job shop processing multi-level assemblies. The performance of the different due date assignment procedures is investigated by a simulation model of the multi-level assembly job shop. The performance measures considered in their study are lead time, lateness, tardiness and percentage tardy.

Roman and Valle (1996) described the problem of reducing the tardiness and percentage of delayed jobs in an assembly shop through a combination of dispatch rule and assignation of due dates. They proposed an assignation rule of due dates based on simulation called TWSIM (Total work based on simulation). The performance measures used in the study are mean flow time, mean tardiness, mean earliness, percentage of tardy jobs, maximum tardiness and standard deviation of the tardiness.

Kim and Kim (1996) considered a short-term production scheduling problem for products with multi-level product structures, with the objective of minimizing the weighted sum of discrete earliness and tardiness of components, subassemblies and final products. The scheduling problem is solved with simulated annealing and genetic algorithms and compared with a commonly used approach called finite loading method.

Mckoy and Egbelu (1998) addressed the problem of scheduling a set of assembled products in a job environment to minimize makespan time. A mathematical model is developed to obtain optimum results and a heuristic algorithm is proposed to solve the problem more effectively. The proposed models are tested and compared on several test problems and the computational results confirmed that the heuristic algorithm stands a much better chance of adoption to the mathematical (MILP) model in a practical environment.

Reeja and Rajendran (2000a) addressed the development of new dispatching rules with a view to address various measures of performance related to fowtime and staging delay of jobs in a dynamic assembly shop environment. They introduced ‘operation synchronization date’ concept in the three dispatching rules. The product structure configuration is considered in three classes viz single level, two level and three level structures. The existing and proposed dispatching rules were evaluated by simulation study and the results indicate that the proposed rules are superior to the existing ones for most measures of performance.

Reeja and Rajendran (2000b) proposed various dispatching rules for scheduling in job shops manufacturing multi-level assembly jobs with the performance measures related to tardiness and earliness. They defined `operation due date' in the context of assembly jobs and used it in the development of dispatching rules. The existing and proposed rules were tested by simulating a hypothetical open assembly shop on three distinct job structure sets to determine the sensitivity of performance of dispatching rules to various job structures.

Mohanasundaram et al. (2002) proposed new dispatching rules with the objectives of minimizing the maximum and standard deviation of flowtime, staging delay, conditional tardiness and absolute lateness for scheduling dynamic assembly job shops. They evaluated the performance of the four proposed rules with the exiting rules using an extensive simulation based investigation by randomly generating jobs with different structures and different shop utilization levels.

Pongcharoen et al. (2002) developed a genetic algorithm-based scheduling tool (GAST) for the scheduling of complex products with deep product structure and multiple resource constraints. Their proposed GA includes a repair process that identifies and corrects infeasible schedules. The algorithm takes account of the requirement to minimize the penalties due to both the early and late delivery of final products while simultaneously considering capacity utilization. Their work investigated appropriate levels for genetic algorithm parameters that produced good schedules with minimum total penalty costs.

Thiagarajan and Rajendran (2003) described the problem of scheduling dynamic assembly shops that manufacture multi-level jobs. They modified some existing dispatching rules and also proposed a pair of new dispatching rules with the consideration of different holding and tardiness cost for different jobs. The proposed rules are evaluated with respect to a number of performance measures such as the weighted mean scheduling cost, the maximum and variance of total scheduling cost, weighted mean flowtime and weighted mean tardiness by means of different levels of shop utilization and job due-date settings

Jang et al. (2003) presented a mixed integer programming model for flexible job shop scheduling problem with multi-level jobs considering complex routing and alternative machines. They developed a genetic algorithm with a new gene design to represent machine assignment, operation sequences and relative level of the operation to the final assembly operation for the problem with minimization of tardiness as the performance criterion. The proposed algorithm is compared with several dispatching rules and proved its competence in different problem instances.

Pongcharoen et al. (2004) described the development of a genetic algorithm for assembly job shop problem with multiple levels of product structure and multiple resource constraints. A repair processes was included in GA to rectify infeasible schedules that may be produced during evolution process. The GA was applied to the data from a company that manufactures capital goods and the schedules demonstrated a large reduction in tardiness and earliness costs when compared with those obtained from a company employing a traditional scheduling method. They performed a statistical analysis on factorial experiment and showed that problem size, the number of generations, the population size and the probability of mutation were statistically significant.

Lagodimos et al. (2004) developed a scheduling procedure for a multistage fabrication shop in a manufacturing plant producing commercial refrigeratior components so as to ensure continuous operation of assembly stations. They proposed an algorithm that applies general planning principles adapted to the needs of the environment under consideration and makes use of existing heuristic rules for arriving at sequencing decisions. Their proposed algorithm outperformed previous company scheduling practice.

Thiagarajan and Rajendran (2005) carried out an investigation of dispatching rules for scheduling in dynamic assembly job-shops with the consideration of different weights for earliness, tardiness and flowtime of jobs. They conducted simulation studies for assembly job-shops that manufacture different types of multi-level jobs, with different levels of shop utilization and job due-date settings. The proposed dispatching rules were evaluated with respect to a number of measures of performance and their efficiency for scheduling assembly job shops were proved.

Guo et al. (2006) developed a universal mathematical model for the job shop scheduling problem in a mixed and multi-product assembly environment based on an apparel industry. A genetic optimization process is proposed by them to solve the model which includes a new chromosome representation, a heuristic initialization process and modified crossover and mutation operators. The objective pursued is to minimize the total penalties of earliness and tardiness. The effectiveness of the proposed method is evaluated using data obtained from the apparel industry.

Hicks and Pongcharoen (2006) investigated the use of dispatching rules in stochastic situation using data obtained from a capital goods company that produce three families of complex products. They explored the effect of four operational parameters such as minimum set-up, processing and transfer times and the data update period on manufacturing performance under infinite capacity conditions with stochastic processing times. The significance of these factors and the relative performance of eight dispatching rules in terms of mean tardiness with finite capacity and stochastic processing times were investigated and reported.

Pathumnakal and Egbelu (2006) described the problem of scheduling assembly job shops in which each product consists of components and subassemblies and each component or subassembly may need additional processing before it can mate with other parts. The objective is to minimize the weighted earliness penalty subject to no products being tardy. A heuristic is developed to generate optimal or near-optimal solution to the problem. The efficiency of the proposed heuristic is demonstrated by the results obtained from the test problems.

Dimyati (2007) addressed a problem of scheduling in a make-to-order job shop with product assembly consideration. A mixed integer linear programming model is developed to solve the model in which the objective is to minimize makespan time. A heuristic algorithm is also proposed for the model to obtain optimal or near optimal solution in a reasonable computational time.

Natarajan et al. (2007) investigated the problem of scheduling in assembly job shops with jobs having different weights for holding and tardiness. They proposed new priority dispatching rules that minimize the performance measures related to weighted flow time and weighted tardiness of jobs. They are also modified existing unweighted dispatching rules in view of the consideration of weights for flow time and tardiness of jobs. The performances of the modified and proposed dispatching rules are compared through simulation experiments with the consideration of a number of different experimental settings involving due-date setting, utilization levels and types of job structures.

Chan et al. (2008a) combined lot streaming (LS) and assembly job shop scheduling problem (AJSP), extending LS applicability to both machining and assembly. They proposed a genetic algorithm to determine sub lot combinations and simple dispatching rules to solve AJSP with all sub-lots. The objective pursued in the developed model is to minimize late cost and inventory cost. A number of test problems are examined to study the impact of LS on AJSP and the experimental results confirmed that equal size LS outperforms varied size LS with respect to the objective function.

Chan et al. (2008b) studied lot streaming to assembly job shop scheduling problem with part sharing among distinct products. An evolutionary approach with genetic algorithm is proposed to solve the problem. The proposed system measurements are makespan, inventory cost, setup cost and WIP. The computational results proved the efficiency of the proposed algorithm.

Girish (2009) addressed an assembly job shop scheduling problem with multiple routings. He proposed three population based search heuristics, a genetic algorithm, an ant colony optimization algorithm and a particle swarm optimization algorithm for the problem. The objective is to minimize total tardiness cost. The efficiency of the proposed algorithms are demonstrated by the results obtained from the test problems.

Huang and Lu (2009) proposed a bilevel programming approach for assembly job shop scheduling. Two levels of decision makers are identified in the model. The first level aims to minimize the earliness and tardiness of completed jobs and the second level aims to minimize the average shop floor throughput time. Their work identifies the best choice for the project manager under different job shop utilization levels using a simulation approach.

Wong et al. (2009) developed an innovative GA based approach to solve resource-constrained assembly job shop scheduling problem with lot streaming technique. The primary objective of their model is the minimization of total lateness cost of all final products. In addition, two more experimental factors were introduced namely common part ratio and workload index. Their study provides some useful insights about the application of lot streaming technique to resource-constrained assembly job shop scheduling problem.

Gomes et al. (2009) described a problem of scheduling flexible job shop with recirculation and assembly. They developed two mixed integer linear programming models and solved the problems using a due-date-based objective function. The models conveyed discrete and continuous approaches both in the modelling of time as well as in the assignment of jobs to machines. The proposed methods are tested for their performance in an assembly job shop environment of a mould making industry.

Omkumar et al. (2009) presented a static assembly job shop scheduling problem with the objective of minimizing makespan. They proposed a genetic algorithm for solving the problem. The performance of the proposed GA is evaluated with some of the best performing dispatching rules used for scheduling dynamic assembly job shops. The product structures considered are single level, two level and three level structures. The proposed GA outperformed dispatch rules for all the three levels considered.

Lu et al. (2010) described an integrated application of order review/release (ORR) mechanism and dispatching rules to solve assembly job shop scheduling problems. They conducted the full factorial experiment for a simulated assembly job shop and the ability of different combinations of ORR-dispatching rules in optimizing due date and flow time related performance measures is evaluated.

The literature survey reveals that the problem of assembly job shop scheduling has been least researched even though most of the products manufactured in industries consists of both processing and assembly operations. The scheduling of assembly job shops with multiple objectives are seldom considered by the researchers though it has significant practical interest. However there are some previous studies on job shop scheduling with multiple objectives.

2.3 MULTI-OBJECTIVE JOB SHOP SCHEDULING

The job shop is one of the most common manufacturing environments in the world and scheduling in job shops is an important operational problem. In real-world production environments, scheduling in job shops must be done often to achieve several objectives simultaneously. A comprehensive survey of the literature dealing with multi-objective job shop scheduling problems are included in this section.

Ponnambalam et al. (2001) addressed a classical job shop scheduling problem with multi-objective criterion (minimization of weighted sum of makespan time, total tardiness and total idle time of all machine). A multi-objective genetic algorithm that adopts Giffler and Thompson (GT) procedure for actives feasible schedule generation is proposed to solve the problem. The performance test and validation of the proposed algorithm is also discussed.

Kacem et al. (2002a) presented two approaches to solve multi-objective flexible job shop scheduling problems. The first one is the approach by localization to solve the problem of resource allocation and builds an ideal assignment model and the second one is an evolutionary approach controlled by the assignment model. The proposed optimization method was based on minimization of the makespan and balancing of the workloads of machines. They explained some of the practical and theoretical considerations in the construction of a more robust encoding of the genetic algorithms to solve the flexible job-shop problem.

Kacem et al. (2002b) addressed a multi-objective flexible job-shop scheduling problem with the objective of simultaneously minimizing the makespan, the total workload of machines and the workload of the most loaded machine. A Pareto based hybrid approach is developed to solve the problem. The proposed algorithm exploits the knowledge representation capabilities of fuzzy logic and the adaptive capabilities of evolutionary algorithms and its efficiency is illustrated through different examples.

Hsu et al. (2002) developed a multi-objective evolutionary algorithm that exploits NSGA II (non dominated sorting genetic algorithm II) for solving flexible job shop scheduling problems. They proposed a set of mutation heuristics in a view to direct the mutation towards best solutions. The objectives pursued are to minimize the makespan, the total workload of machines and the critical machine workload simultaneously. Computational experiments proved the efficiency of the proposed approach.

Baykasoglu et al. (2004) presented a practical modeling and solution approach which makes use of grammars, multi-objective tabu search and a dispatching rule based heuristics for multi-objective flexible job shop problems. The objectives selected are makespan, total tardiness and load balance. The effectiveness of the proposed approach is evaluated with the help of an example problem.

Miragliotta and Perona (2005) introduced RESDES (reentrant shop decentralised scheduling) approach to the scheduling of reentrant shops. The new heuristic based approach adopted the job shop BAL rule (balanced evaluation of the shop oriented needs), which implements a multi-objective approach, in that decisions are made on the basis of a combination of aspects relating both to the job oriented (measured through the mean tardiness) and shop oriented (measured through the cumulated throughput) domains. An extensive computational experiment based on two case studies, belonging to semiconductors and metalworking businesses is presented to highlight robustness and real life suitability of the proposed approach.

Xia and Wu (2005) discussed a hybrid approach for the multi-objective flexible job-shop scheduling problem. The proposed approach makes use of particle swarm optimization to assign operations on machines and simulated annealing algorithm to schedule operations on each machine. The objectives pursued are to minimize the makespan, the total workload of machines, and the workload of the critical machine simultaneously. The computational results showed the effectiveness of the proposed algorithm.

Lei and Wu (2006) described a multi-objective job shop scheduling problem. They designed a crowding measure based multi-objective evolutionary algorithm, which makes use of the crowding measure to adjust the external population and assign different fitness for individuals. The objective is to minimize the makespan and the total tardiness of jobs simultaneously. The C metric is used for comparison and the computational results demonstrated the usefulness of the proposed algorithm.

Liu et al. (2006) introduced a variable neighborhood particle swarm optimization algorithm (VNPSO), consisting of a combination of the variable neighborhood search and particle swarm optimization for solving the multi-objective flexible job shop scheduling problems. The objective is to minimize the makespan and the flowtime simultaneously. The effectiveness and performance of the proposed algorithm is illustrated with three representative instances based on practical data.

Lei and Xiong (2007) addressed a multi-objective job shop scheduling problem with stochastic processing time. The objective is to simultaneously minimize the expected makespan and the expected total tardiness. A multi-objective evolutionary algorithm is presented that uses a permutation-based representation method and the corresponding encoding procedure. A crowding measure is used to maintain the external archive and assign fitness in the proposed algorithm. The computational results demonstrated the performance of the algorithm in stochastic job shop scheduling.

Loukil et al. (2007) addressed a multi-objective flexible job shop scheduling problem with particular constraints: batch production, existence of two steps: production of several sub-products followed by the assembly of the final product, possible overlaps for the processing periods of two successive operations of a same job. They developed a multi-objective simulated annealing approach for solving the problem. The different objectives considered are the minimization of the makespan, the mean completion time, the maximal tardiness and the mean tardiness. Their research is based on a real case study, concerning a Tunisian firm.

Liu et al. (2007) proposed a hybrid algorithm, the variable neighborhood particle swarm optimization (VNPSO) for solving the multi-objective flexible job-shop scheduling problems. The problem is to determine an assignment and a sequence of the operations on all machines that simultaneously minimize the flowtime and the makespan. The empirical results indicated the efficiency the proposed algorithm, especially for large scale problems.

Jia et al. (2007a) developed a hybrid particle swarm optimization algorithm to solve the multi-objective flexible job shop scheduling problem, which integrates the global search ability of PSO and the superiority of escaping from a local optimum with chaos. The objective considered is to minimize the makespan, the total workload of the machines and the workload of the most loaded machine. Computational experiments with typical problem instances are conducted to evaluate the performance of the proposed method and proved its effectiveness in terms of the quality of solutions and computational time.

Jia et al. (2007b) addressed a multi-objective flexible job-shop problem with the objective simultaneously minimizing the makespan and the maximum lateness. A Pareto based multi-objective particle swarm algorithm is proposed to solve the problem. To increase the diversity of the population and overcome the premature convergence problem, mutation operators are introduced into the proposed algorithm. The performance of the proposed algorithm is demonstrated with different benchmark instances and proved its efficiency.

Lei (2008) presented Pareto archive particle swarm optimization (PAPSO) for multi-objective job shop scheduling problem. The objective is to simultaneously minimize makespan and total tardiness of jobs. PAPSO combined the global best position selection with archive maintenance. The effectiveness of PAPSO was tested on a set of benchmark problems and the computational results showed that the algorithm is capable of producing a number of high-quality Pareto optimal scheduling plans.

Tay and Ho (2008) proposed a genetic programming (GP) based approach for discovering effective composite dispatching rules for solving the multi-objective flexible job-shop problems. The experimental results showed that composite dispatching rules generated by genetic programming framework outperforms the single dispatching rules with respect to minimum makespan, mean tardiness, and mean flow time objectives.

Xing et al. (2008) presented a simulation model to solve the multi-objective flexible job shop scheduling problems with objectives of minimizing makespan, the total workload of machines and critical machine workload. They validated the model by five representative instances based on practical data.

Manikas and Chang (2008) demonstrated that genetic algorithms (GA) can be used to produce near to optimal solutions for a variety of production optimization criteria in a job shop environment that includes sequence-dependent setup times. The performance measures considered are earliness, tardiness, job rank and makespan time. They showed that the proposed GA performance is relatively insensitive to the problem data and criteria for evaluation.

Vilcot and Billaut (2008) investigated a bicriteria general job shop scheduling problem from printing and boarding industry with the objectives of minimizing the makespan and the maximum lateness. They proposed a genetic algorithm based on NSGA II for solving the problem. The initial population of this algorithm is either randomly generated or partially generated by using a tabu search algorithm that minimizes a linear combination of the two criteria. The algorithms are tested on various benchmark instances derived from the literature. The results illustrated the interest of introducing the solutions of the tabu search algorithm into the initial population of the genetic algorithm, both in terms of solutions quality and of computation time.

Fattahi (2009) proposed a Pareto approach to solve the multi objective flexible job shop scheduling problems. The objectives considered are to simultaneously minimize the makespan and total weighted tardiness. An effective simulated annealing algorithm based on the proposed approach is presented to solve the problem. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the solution process. Numerical experiments showed that the proposed algorithm is capable to obtain a set of Pareto optimal solutions in a small time.

Zhang et al. (2009) proposed a hybrid particle swarm optimization algorithm by combining PSO and tabu search to solve the multi-objective flexible job-shop scheduling problems with several conflicting and incommensurable objectives. The proposed algorithm which integrates local search and global search scheme possesses high search efficiency. The objectives considered are to minimize the makespan, the maximal machine workload and the total workload of machines simultaneously. They combined the different objectives into a weighted sum. The obtained computational results demonstrated the effectiveness of the proposed approach.

Sha and Lin (2010) constructed a particle swarm optimization for an elaborate job-shop scheduling problem with multiple objectives including minimization of makespan, machine idle time, and total tardiness. They proposed a new particle position representation, particle movement, and particle velocity to incorporate the discrete solution spaces of scheduling optimization problems. The maintenance of Pareto optima and a diversification procedure are used in the algorithm to achieve better performance. The proposed algorithm was used to solve various benchmark problems and the results demonstrated that the algorithm performed better in search quality and efficiency than traditional evolutionary heuristics.

Wang et al. (2010) presented a multi-objective genetic algorithm (MOGA) based on immune and entropy principle to solve the multi-objective flexible job shop scheduling problem with objectives of minimizing makespan, the total workload of machines and critical machine workload. They used an immune and entropy principle to keep the diversity of individuals and overcome the problem of premature convergence. They applied a fitness scheme based on Pareto-optimality and efficient crossover and mutation operators to adapt to the special chromosome structure. The MOGA is validated with the help of some representative instances of the problem.

Tavakkoli-Moghaddam et al. (2011a) proposed a Pareto archive PSO combined with genetic operators and variable neighborhood search for solving a multi-objective job shop with respect to the mean weighted completion time and the sum of the weighted tardiness/earliness costs. A number of test problems are solved to validate the proposed PSO. The reliability of the proposed algorithm is evaluated with the assistance of three comparison metrics, such as quality metric, spacing metric, and diversity metric. The results indicated that the proposed PSO outperformed NSGA-II in various problem instances.

Tavakkoli-Moghaddam et al. (2011b) presented a new hybrid Pareto archive particle swarm optimization (PSO) algorithm for solving a bi-objective job shop scheduling problem with respect to the mean weighted flow time and the sum of the weighted tardiness and earliness costs. The comparison metrics such as quality metric, spacing metric and diversity metric were applied to validate the efficiency of the proposed hybrid PSO. Experimental results showed the performance of the proposed algorithm in terms of the solution quality and diversity level.

Li et al. (2011) developed a hybrid Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problem. The objectives pursued are to minimize the makespan, the total workload and the critical machine workload simultaneously. In the proposed algorithm, each solution corresponds to a food source is composed of two components, i.e., the routing component and the scheduling component. A well-designed crossover operator is developed for the employed bees to learn valuable information from each other. A fast Pareto set update function and an external Pareto archive set are introduced in the algorithm. Two local search approaches are incorporated to balance the exploration and exploitation capability of the algorithm.

Kachitvichyanukul and Sitthitham (2011) proposed a two-stage genetic algorithm (2S-GA) for multi-objective job shop scheduling problems with the objective of minimizing makespan, total weighted earliness, and total weighted tardiness. They applied a parallel GA in stage 1 to evolve population of high quality solutions for each objective function and the combined populations are used as an initial population in stage 2 steady-state GA. The evolution process of stage 2 is based on the weighted aggregating function as fitness function. They validated the effectiveness of the proposed algorithm with different benchmark instances.

Lei (2011) developed a simplified multi-objective genetic algorithm (SMGA) for the stochastic job shop scheduling problem with exponential processing time to minimize makespan and total tardiness ratio simultaneously. In the proposed algorithm, a novel crossover, binary tournament selection based on rank and the weighted objective and a simplified external archive updating strategy are adopted. They showed the good performance of SMGA on the problem with computational experiments.

Li et al. (2012) proposed a hybrid shuffled frog-leaping algorithm for solving the multi-objective flexible job shop scheduling problem with the objective of simultaneously minimizing makespan, the total workload of all machines, and the workload of the critical machine. They introduced a new crossover operator in the proposed algorithm and designed several neighborhood structures to direct the local search to more promising search space. The proposed algorithm is tested with different benchmark instances and proved its efficiency.

2.4 STATE OF THE ART

Tables 2.1 and 2.2 present summary of literature review on assembly job shop scheduling (AJS) and multi-objective job shop scheduling (MOJS), respectively.

Table 2.1 Summary of literature review on AJS

Author	Year	Objectives	Methodology
Sculli	1980, 1987	Minimization of flow time, and tardiness measures	Priority dispatching rules
Adam et al.	1987	Minimization of tardiness	Priority dispatching rules
Fry et al.	1989	Minimization of flow time, and tardiness measures	Priority dispatching rules
Philipoom et al.	1989	Minimization of tardiness	Priority dispatching rules
Philipoom et al.	1991	Minimization of flow time and tardiness measures	Priority dispatching rules
Doctor et al.	1993	Maximization of machine utilization with the objective of satisfying due dates	Heuristic procedures based on priority dispatching rules
Adam et al.	1993	Minimization of lead time and tardiness measures	Priority dispatching rules
Roman and Valle	1996	Minimization of flow time and tardiness measures	Priority dispatching rules and a heuristic
Kim and Kim	1996	Minimization of weighted sum of discrete earliness and tardiness penalties	Simulated annealing and genetic algorithm
McKoy and Egbelu	1998	Minimization of flow time	MILP model and heuristic algorithm
Reeja and Rajendran	2000a, 2000b	Minimization of flow time and staging delay	Priority dispatching rules

Table 2.1 (Continued)

Author	Year	Objectives	Methodology
Mohanasundaram et al.	2002	Minimization of flow time based measures and tardiness	Priority dispatching rules
Pongcharon et al.	2002, 2004	Minimization of penalties caused by early and late delivery of components, assemblies and final products	Genetic algorithm
Thiagarajan and Rajendran	2003	Minimization of total scheduling cost consisting of the sum of holding and tardiness costs	Priority dispatching rules
Jang et al.	2003	Minimization of tardiness	Genetic algorithm
Lagodimos et al.	2004	Minimization of total makespan time of jobs in different lots	Algorithm based on heuristic rules
Thiagarajan and Rajendran	2005	Minimization of sum of weighted earliness, weighted tardiness and weighted flow time of jobs	Priority dispatching rules
Guo et al.	2006	Minimization of total penalty of earliness and tardiness	Genetic algorithm
Hicks and Pongcharoen	2006	Minimization of mean tardiness	Priority dispatching rules
Pathumnakul and Egbelu	2006	Minimization of earliness penalty subject to no products being tardy and no cost for machines staying idle	Heuristics methods based on priority dispatching rules
Dimyati	2007	Minimization of makespan	Mixed integer linear programming model and a heuristic algorithm
Natarajan et al.	2007	Minimization of weighted flow time and weighted tardiness of jobs	Priority dispatching rules
Chan et al.	2008a, 2008b	Minimization of makespan, inventory cost, setup cost and WIP	Genetic algorithm
Girish	2009	Minimization of total tardiness cost	Three population based search heuristics

Table 2.1 (Continued)

Author	Year	Objectives	Methodology
Huang and Lu	2009	Minimization of earliness and tardiness of completed jobs and the average shop floor throughput time	Bilevel programming approach
Wong et al.	2009	Minimization of total lateness cost	Genetic algorithm
Gomes et al.	2009	Minimization of due-date-based performance measure	Mixed integer linear programming
Omkumar et al.	2009	Minimization of makespan	Genetic algorithm
Lu et al.	2010	Minimization of due date and flow time related performance measures	Priority dispatching rules

Table 2.2 Summary of literature review on MOJS

Author	Year	Objectives	Methodology
Ponnambalam et al.	2001	Simultaneously minimizing makespan time, total tardiness and total idle time of all machine	Genetic algorithm (Weighted approach)
Kacem et al.	2002a	Simultaneously minimizing makespan and balancing of the workloads of machines	Approach by localization and genetic algorithm (Non Pareto approach)
Kacem et al.	2002b	Simultaneously minimizing makespan, total workload of machines and workload of the most loaded machine	Hybridization of fuzzy logic and evolutionary algorithm (Pareto approach)
Hsu et al.	2002	Simultaneously minimizing makespan, the total workload of machines and the critical machine workload	Multi-objective evolutionary algorithm (Pareto approach)
Baykasoglu et al.	2004	Simultaneously minimizing makespan, total tardiness and load balance	Grammars, multi-objective tabu search and a dispatching rule based heuristics (Pareto approach)

Table 2.2 (Continued)

Author	Year	Objectives	Methodology
Miragliotta and Perona	2005	Simultaneously minimizing mean tardiness and cumulated throughput	RESDES approach (Weighted approach)
Xia and Wu	2005	Simultaneously minimizing makespan, the total workload of machines, and the workload of the critical machine	Hybrid particle swarm optimization - simulated annealing (Weighted approach)
Lei and Wu	2006	Simultaneously minimizing makespan and total tardiness of jobs	Crowding-measure-based multiobjective evolutionary algorithm (Pareto approach)
Liu et al.	2006	Simultaneously minimizing makespan and sum of completion time	Variable neighborhood particle swarm optimization (Weighted approach)
Lei and Xiong	2007	Simultaneously minimizing expected makespan and expected total tardiness	Multi-objective evolutionary algorithm (Pareto approach)
Loukil et al.	2007	Simultaneously minimizing makespan, mean completion time, maximal tardiness and mean tardiness	Simulated annealing (Weighted approach)
Liu et al.	2007	Simultaneously minimizing flowtime and makespan	Variable neighborhood particle swarm optimization (Weighted approach)
Jia et al.	2007a	Simultaneously minimizing makespan, total workload of the machines and workload of the most loaded machine	Hybrid particle swarm optimization algorithm (Weighted approach)
Jia et al.	2007b	Simultaneously minimizing makespan and maximum lateness	Multi-objective fully informed particle swarm algorithm (Pareto approach)
Lei	2008	Simultaneously minimizing makespan and total tardiness of jobs	Pareto archive particle swarm optimization (Pareto approach)

Table 2.2 (Continued)

Author	Year	Objectives	Methodology
Tay and Ho	2008	Simultaneously minimizing makespan, mean tardiness, and mean flow time	Genetic programming based approach for discovering effective composite dispatching rules (Weighted approach)
Xing et al.	2008	Simultaneously minimizing makespan, the total workload of machines and critical machine workload	Simulation model (Weighted approach)
Manikas and Chang	2008	Simultaneously minimizing earliness, tardiness, job rank and makespan	Genetic algorithm (Weighted approach)
Vilcot and Billaut	2008	Simultaneously minimizing makespan and maximum lateness	Genetic algorithm based on NSGA II (Pareto approach)
Fattahi	2009	Simultaneously minimizing makespan and total weighted tardiness	Hybrid algorithm based on simulated annealing (Pareto approach)
Zhang et al.	2009	Simultaneously minimizing makespan, the maximal machine workload and the total workload of machines	Hybrid particle swarm optimization – tabu search (Weighted approach)
Sha and Lin	2010	Simultaneously minimizing makespan, machine idle time, and total tardiness	Particle swarm optimization (Pareto approach)
Wang et al.	2010	Simultaneously minimizing makespan, the total workload of machines and critical machine workload	Multi-objective genetic algorithm (Pareto approach)
Tavakkoli-Moghaddam et al.	2011a	Simultaneously minimizing mean weighted completion time and the sum of the weighted tardiness/earliness costs	Particle swarm optimization with genetic operators and variable neighborhood search (Pareto approach)
Tavakkoli-Moghaddam et al.	2011b	Simultaneously minimizing mean weighted flow time and the sum of the weighted tardiness and earliness costs	Hybrid Pareto archive particle swarm optimization algorithm (Pareto approach)

Table 2.2 (Continued)

Author	Year	Objectives	Methodology
Li et al.	2011	Simultaneously minimizing makespan, total workload and critical machine workload	Pareto-based discrete artificial bee colony algorithm (Pareto approach)
Kachitvichyanukul and Sitthitham	2011	Simultaneously minimizing makespan, total weighted earliness, and total weighted tardiness	Two-stage genetic algorithm
Lei	2011	Simultaneously minimizing makespan and total tardiness ratio	Simplified multi-objective genetic algorithm (Pareto approach)
Li et al.	2012	Simultaneously minimizing makespan, total workload and critical machine workload	Hybrid shuffled frog-leaping algorithm (Pareto approach)

The literature review on assembly job shop scheduling and multi-objective job shop scheduling reveals the following:

- The quantum of research in scheduling of assembly job shop involving machining operations is very limited.
- Optimization techniques ruled out for solving assembly job shop scheduling problems due to the complex nature of the problem.
- Approximation algorithms have been identified as a useful tool for assembly job shop scheduling problems.
- The priority dispatching rules do not guarantee optimal performance in varying environment.
- Metaheuristic techniques have emerged as powerful tool for multi-objective optimization.
- Research on multi-objective job shop scheduling is relatively less compared to single objective job shop scheduling.

- Research on assembly job shop scheduling has tended to concentrate on a single criterion to optimize even though most of the industries require multi-objective optimization.

The above discussions indicate that there is a lot of scope open for doing research in this area. A few of them are:

- Modeling multi-objective assembly job shop scheduling problems based on weighted approach and Pareto approach.
- Application of metaheuristics such as genetic algorithm and tabu search in multi-objective assembly job shop scheduling.
- Development of hybrid optimization strategy for solving multi-objective assembly job shop scheduling.

2.5 SUMMARY

In this chapter the outcome of an exhaustive literature survey on assembly job shop scheduling and multi-objective job shop scheduling are reported. The description of the problem under consideration is included in Chapter 3.

CHAPTER 3

PROBLEM DESCRIPTION

3.1 INTRODUCTION

This chapter addresses an assembly job shop scheduling problem with multiple objectives. This combinatorial optimization problem is termed as multi-objective assembly job shop scheduling problem (MOAJSP). The problem environment, assumptions, objective criterions, problem definition, mathematical formulation and approaches considered in this thesis for multi-objective optimization are presented in the following sections.

3.2 PROBLEM ENVIRONMENT

The problem environment consists of an assembly job shop comprising of a machine shop and an assembly shop. The components required for manufacturing a set of N products are first processed in machine shop and then assembled in the assembly shop to form complete end products. The sub-assemblies may also require processing on machines before final product assembly. Thus completing the end products implies completing all the processing and assembly requirements of all components and assemblies of a product and overall N products. The general description of the problem environment is given below:

- There are M machines in the system and N products to be manufactured.
- Each product p requires a set of components (CS_p) to be made. Each component i ($i \in CS_p$) requires J_{pi} number of precedence-constrained operations to be performed.

- Each product p requires a set of assembling operations (AS_p) to be performed and it can be started only when its preceding operations in the set E_{pij} are completed.
- Each assembly or sub-assembly i ($i \in AS_p$) may also require J_{pi} number of precedence-constrained processing operations to be performed subsequent to the assembly operations.
- Each operation O_{pij} should be processed/assembled on particular machine with the processing/assembling time t_{pij}.
- The final operation forms the complete product. The examples of product structures are given in Figures 3.1 to 3.3.

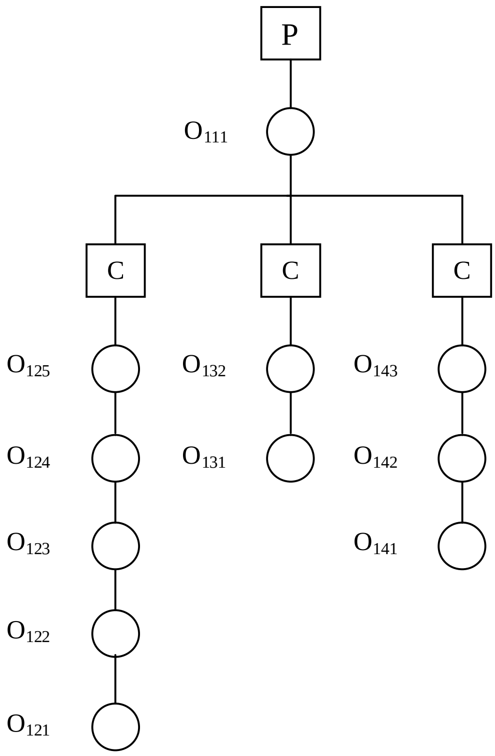

Figure 3.1 Product 1: Flat structure

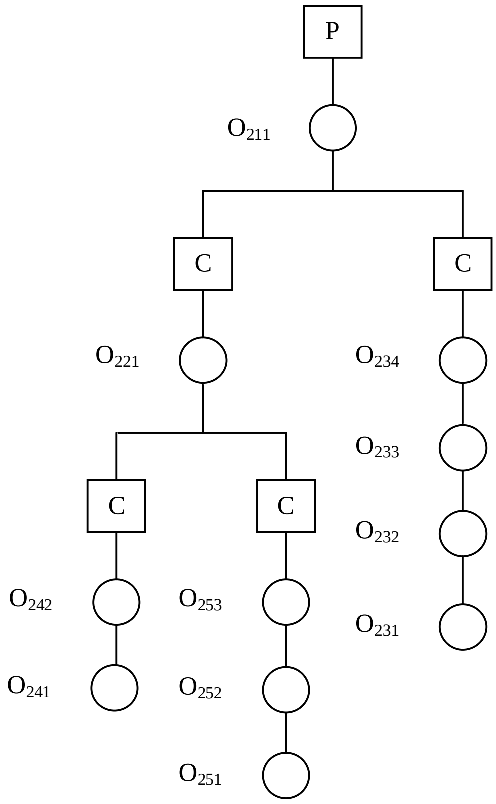

Figure 3.2 Product 2: Tall structure

3.3 ASSUMPTIONS

The following assumptions are made in the study:

- Components are independent and consist of strictly ordered operation sequences and no priorities are assigned to any component type.
- Setup time is independent of operation sequence and is included in the processing time.
- Transportation time is negligible.
- Machines are independent of each other.
- A machine cannot process more than one job at a time.
- No components may be processed on more than one machine at a time.
- All components are simultaneously available at time zero.
- Outsourced components are available at the time of assembly.

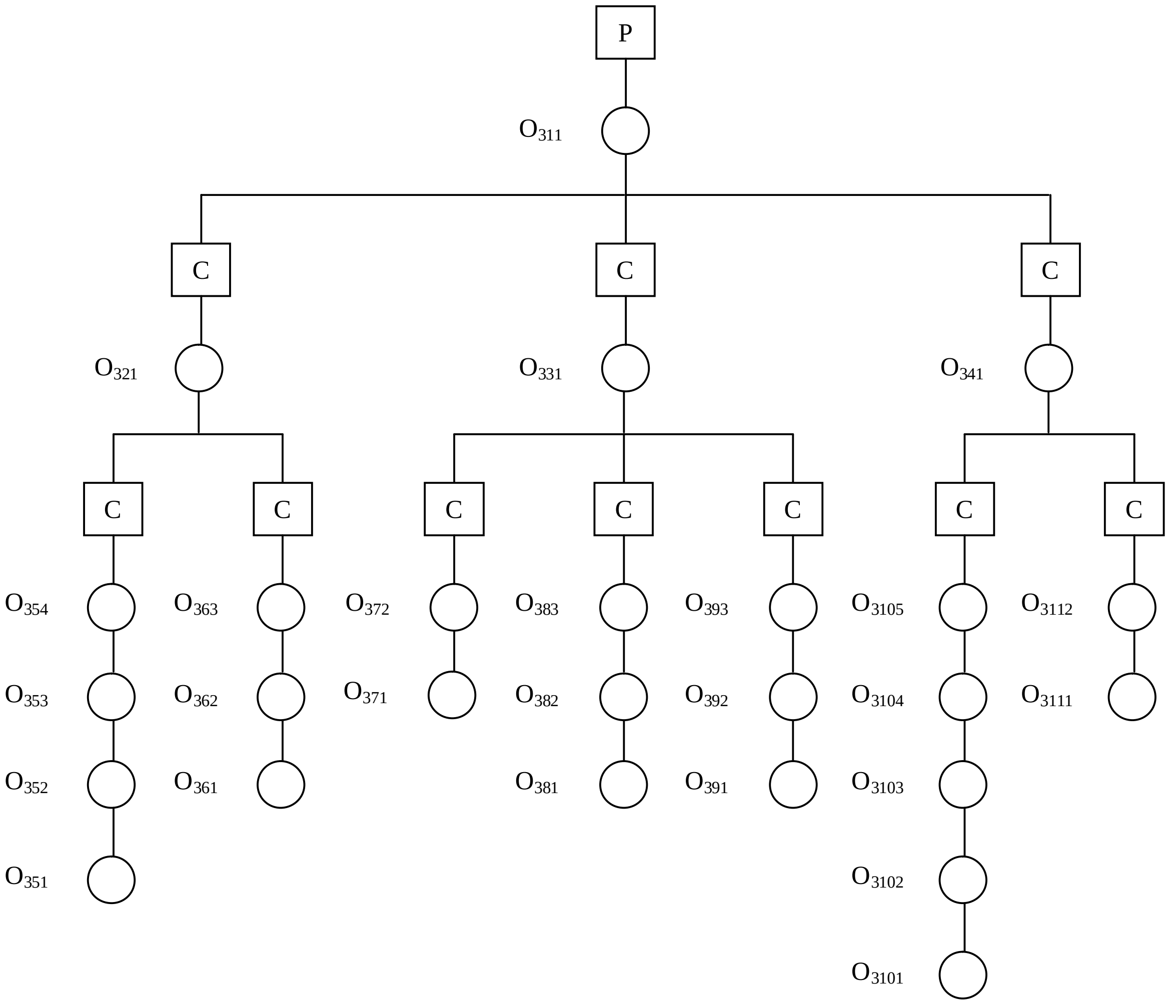

Figure 3.3 Product 3: Complex structure

3.4 OBJECTIVES

Many industries have tradeoffs in their scheduling problems where multiple objectives need to be considered in order to optimize the overall performance of the system (Fattahi, 2009). For reflecting the real world situation effectively, a multi-objective performance measure is applied as the objective function in this study to construct schedules. The minimization of cost and the maximization of customer satisfaction are two main foci of the practical application (Lei, 2008). A completion-time-related objective such as makespan aims to reduce production time and increase facility utilization which is a critical factor towards minimizing cost and a due-date-related objective such as total tardiness aims to meet on-time delivery which is a critical factor towards realizing customer satisfaction. Therefore the objective considered in this thesis is to simultaneously minimize the following performance measures:

1. Makespan = max $\{F_p\}$ where F_p is the completion time of product p.
2. Total tardiness of products = $\sum$ {max $[0, L_p]$} where L_p is the lateness of product p

3.5 PROBLEM DEFINITION

The problem can be defined as:

"Determination of optimal or near optimal schedules for the multi-objective assembly job shop scheduling problem

- by sequencing the processing and assembly operations on the machines for the objective of minimization of makespan time and total tardiness simultaneously.
- given the processing time of all operations, the assembly time for all assembly operations, the precedence constraints of all components and subassemblies along with product due dates."

3.6 MATHEMATICAL FORMULATION

A mathematical model (adopted with modification from Girish, 2009) that sequences the set of N products over a set of M machines is developed. The notations used are:

N	Number of products
M	Number of machines
p, p'	Index for products ($p=1, 2, \ldots\ldots, N$)

i, i'	Index for components ($i=1, 2, \ldots\ldots., n_p$)
j, j'	Index for operations ($j=1, 2, \ldots\ldots., J_{pi}$)
m	Index for machines ($m=1, 2, \ldots\ldots., M$)
n_p	Number of components of product p
J_{pi}	Number of processing operations of component i of product p
O_{pij}	Operation j of component i of product p
S_{pij}	Start time of operation O_{pij}
C_{pij}	Completion time of operation O_{pij}
F_p	Finish time of product p
D_p	Due date of product p
t_{pijm}	Processing/assembly time of operation O_{pij} on machine m
L_m	Set of processing operations O_{pij} that can be assigned to machine m
E_{pij}	Set of preceding operations O_{pij} of assembly operation j of product p
T_p	Tardiness of product p
H	A large positive integer
$X_{pijp'i'j'm}$	Decision variable for generating a sequence between operations O_{pij} and $O_{p'i'j'}$ for loading on machine m

Objectives:

$$Min\ Z_1 = [Max\ (F_1, F_2, \ldots \ldots, F_N)] \tag{3.1}$$

$$Min\ Z_2 = \sum_{p=1}^{N} T_p \tag{3.2}$$

Subject to:

$$T_p = Max\left(0, F_p - D_p\right) \forall\, p \tag{3.3}$$

$$F_p = Max\left(C_{111},\ C_{112}, \dots\dots,\ C_{piJ_{pi}}\right)\ \forall\, p \tag{3.4}$$

$$C_{pij} - S_{pij} - t_{pij} = 0\ \forall\,(p,i,j): i = 1,2,\dots\dots, n_p \tag{3.5}$$

$$C_{p'i'j'} - C_{pij} + H\left(1 - X_{pijp'i'j'm}\right) \geq t_{p'i'j'm}, \qquad \forall m, (p,i,j), (p',i',j'):$$

$$i = 1,2,\dots\dots, n_p,\ O_{pij} \in L_m and\ O_{p'i'j'} \in L_m \tag{3.6}$$

$$C_{p'i'j'} - C_{pij} + H.\ X_{pijp'i'j'm} \geq t_{pijm}, \qquad \forall m, (p,i,j), (p',i',j'):$$

$$i = 1,2,\dots\dots, n_p,\ O_{pij} \in L_m and\ O_{p'i'j'} \in L_m \tag{3.7}$$

$$S_{pij} \geq 0\ \forall\,(p,i,j) \tag{3.8}$$

$$S_{pij+1} - C_{pij} \geq 0\ \forall\,(p,i,j): i = 1,2,\dots\dots, n_p\ and\ j = 1,2,\dots\dots, J_i - 1 \tag{3.9}$$

$$S_{pij} - C_{pi'j'} \geq 0\ \forall\,(p,i,j),(p,i',j'): O_{pi'j'} \in E_{pij} \tag{3.10}$$

$$X_{pijp'i'j'm} = \begin{cases} 1, & if\ operation\ O_{pij}\ precedes\ O_{p'i'j'}\ on\ machine\ m \\ 0, & otherwise \end{cases} \tag{3.11}$$

The constraint set (3.3) imposes that the tardiness of each product is the difference between the completion time of its final assembly/processing operation and due date. The constraint set (3.4) imposes that the finish time of each product is the maximum of the completion time of all operations performed to manufacture that product. The constraint set (3.5) imposes that the difference between the completion time and starting time of an operation on a component is equal to its processing time on the machine to which it is assigned. Constraint set (3.5) satisfies the assumption that once an operation has started, it cannot be pre-empted until its completion. The disjunctive constraint sets (3.6) and (3.7) ensure that no two operations can be processed simultaneously on the same machine. The constraint (3.6) becomes active when $X_{pijp'i'j'm}$=1 and constraint (3.7) becomes active when $X_{pijp'i'j'm}$=0. Constraint set (3.8) ensures that the start time of an operation (processing/assembly) is always positive. Constraint set (3.9) represents the precedence relationship among various operations of a component. Constraint set (3.10) represents the precedence relationship between the subassemblies and components.

3.7 APPROACHES FOR MULTI-OBJECTIVE OPTIMIZATION

Many approaches have been developed in the domain of multi-objective meta-heuristic optimization (Hsu et al., 2002). In this thesis, the approaches considered for solving multi objective assembly job shop scheduling problems (MOAJSP) are weighted approach (Xing et al., 2008 & Zhang et al., 2009) and Pareto approach (Lei, 2008 & 2011).

3.7.1 Weighted approach

In this approach the multi-objective problem is transformed into a mono-objective problem by combining the different objectives into a weighted sum.

The advantages of using weighted approach to deal with the multi-objective optimization are given below (Xing et al., 2008):

- It is easy for decision makers to understand the weighted approach.
- It is convenient for developers to implement the weighted approach.
- It is available to modify the weight of different objectives for satisfying the requirement of decision makers.

The weighted sum of makespan and total tardiness is taken as objective function in this problem.

$$Min\ F(s) = w_1 \times F_1(s) + w_2 \times F_2(s),\ w_1 + w_2 = 1 \tag{3.12}$$

where $F(s)$ denotes the objective value of schedule s and w_1 and w_2 represents the weights assigned to the two objective functions $F_1(s)$ and $F_2(s)$, respectively. The weights for different objectives implicate the relative importance of each objective to the other and these weights are determined by the empirical experience.

3.7.2 Pareto approach

This approach is directly based on the Pareto optimization concept. It aims at satisfying two goals: first, converge to the Pareto front and second, obtain diversified solutions scattered all over the Pareto front (Hsu et al., 2002). A feasible solution is called Pareto-optimal when it is not dominated by any other solution in the feasible space. A solution x_1 is said to dominate the other solution x_2, if x_1 is no worse than x_2 in all objectives and x_1 is strictly better than x_2 in at least one objective. For a two-objective optimization problem, a solution x_1 is better than x_2 in first objective and while solution x_1 is worse than x_2 in second objective (i.e., x_2 is better than x_1 in second objective), hence it is customary to say that solutions x_1 and x_2 are non-dominated with respect to each other (Deb, 2001). This approach intends to find all the non-dominated solutions of the multi-objective assembly job shop scheduling problem in order to give more choice to the decision maker and which Pareto optimal solution is the best, depends usually on a decision maker.

3.8 SUMMARY

In this chapter the detailed description of the multi-objective assembly job shop scheduling problem under consideration is carried out. The methodologies developed for the problem with weighted approach and Pareto approach are discussed in Chapters 4 and 5 respectively.

CHAPTER 4

ALGORITHMS WITH WEIGHTED APPROACH

4.1 INTRODUCTION

Three metaheuristic algorithms based on genetic algorithm and tabu search with weighted approach for solving multi-objective assembly job shop scheduling problem are discussed in this chapter. The background and description of the proposed algorithms namely multi-objective genetic algorithm (MOGA), multi-objective tabu search (MOTS) and multi-objective hybrid genetic algorithm (MOHA) are presented in the following sections.

4.2 MULTI-OBJECTIVE GENETIC ALGORITHM

4.2.1 Background

Genetic algorithms (GA) are the most studied and applied metaheuristic in the optimization field because GA are very easy to implement in all sort of problems, and usually guarantee good solutions, whatever the type of solution space. The basic principles of this technique were first laid down by Holland (1975). The technique is based upon the theory of evolution, in which the fitness of an individual determines its ability to survive and reproduce (Goldberg, 1989). In a genetic algorithm, a population of chromosomes which encode candidate solutions to an optimization problem evolves toward better solutions. The traditional GA starts from a population of randomly generated chromosomes. During each generation, the fitness of every chromosome in the population is evaluated and a selection operator works together with a crossover operator and a mutation operator to form a new population. Crossover is performed with a probability between two selected chromosomes and generates offspring by combining features of both chromosomes. Crossover is followed by mutation with a probability to modify a chromosome, which prevents premature convergence to local optimum. The new population is then used for the next iteration and generally it terminates when a maximum number of iterations has been reached. Genetic algorithms have been intensively studied and applied in multi-objective combinatorial optimization problems. In particular, there are some adaptations of genetic algorithms on multi-objective job shop scheduling with weighted approach (Ponnambalam et al., 2001, Tay and Ho, 2008, Manikas and Chang, 2008). In this research, a multi-objective genetic algorithm is developed for assembly job shop scheduling problems with weighed approach.

4.2.2 Description of the proposed MOGA

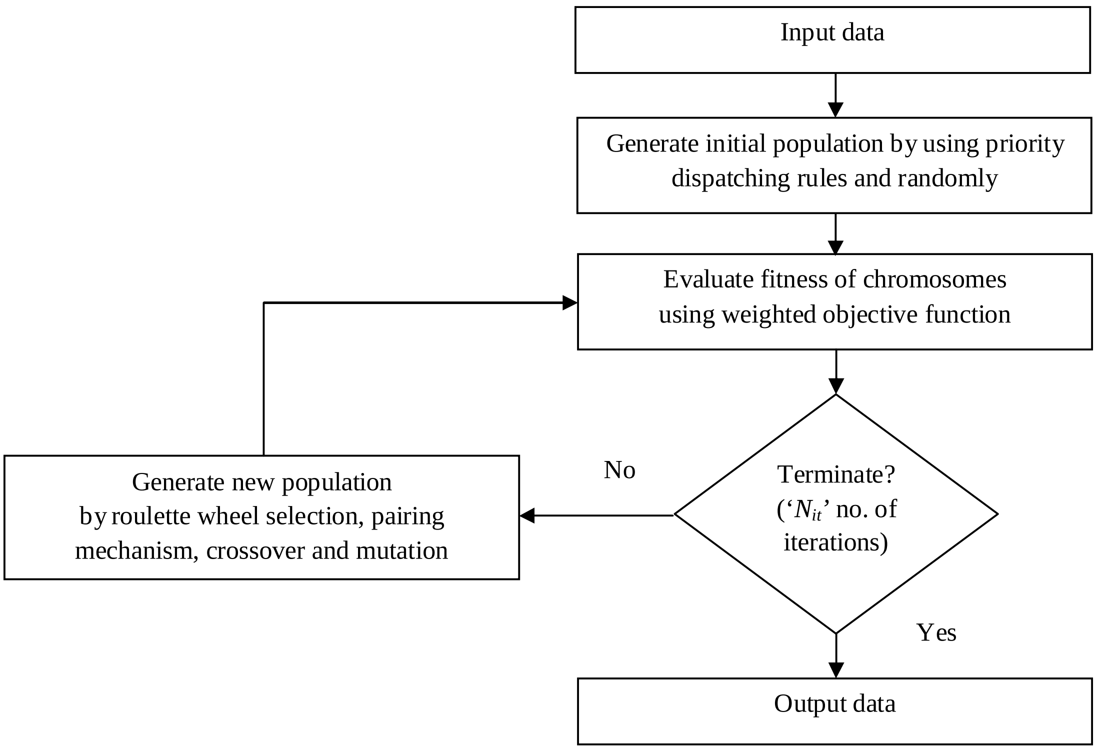

Figure 4.1 Procedure of the proposed MOGA

The procedure of the proposed MOGA is outlined in the flowchart given in Figure 4.1. The detailed description is as follows:

Input data: The following data pertaining to the problem are given as input (Girish, 2009):

- Number of products (N)
- Number of components (n_p) for each product p
- Number of machines in the shop (M)
- Number of processing/assembly operations (J_{pi}) of each component i of product p ($\forall p$)
- Machine number (m) allotted for operation j of each component i of product p ($\forall p$)
- Processing/assembly time of each operation O_{pij} on machine m
- Precedence relationship among various operations of a component of each product p
- Precedence relationship between the subassemblies and components of each product p
- Due date (D_p) for each product p.

The input data requirements for an illustration problem (2 products × 5 machines) are given in Table 4.1. The product structure corresponding to two products are shown in Figures 4.2 and 4.3.

Initial population generation: Initially, a set of chromosomes equal to the size of the population is generated. The first five chromosomes in the initial population are generated from the due-date based priority dispatching rules in the literature, namely, operation due date (ODD), job due date (JDD), total work content remaining (TWKR) (Reeja and Rajendran, 2000b), shortest processing time (SPT) and first-in-first-out (FIFO) (Thiagarajan and Rajendran, 2003). The remaining chromosomes in the initial population are generated randomly.

Table 4.1 Data of the illustration problem

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	16	4	1	1	1	1	5	{(1, 2, 2), (1, 3, 1), (1, 4, 2)}
				2	2	1	3	8	
						2	2	3	
				3	1	1	5	4	
				4	2	1	4	2	
						2	3	1	
2	T	21	5	1	1	1	1	9	{(2, 2, 1), (2, 3, 3)}
				2	1	1	1	3	{(2, 4, 1), (2, 5, 2)}
				3	3	1	2	7	
						2	3	2	
						3	5	1	
				4	1	1	2	5	
				5	2	1	5	6	
						2	4	3	

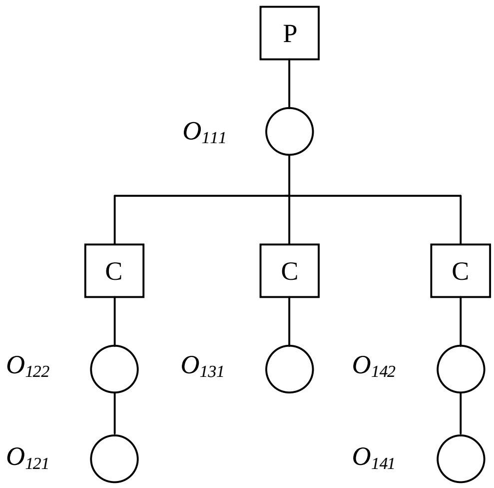

Figure 4.2 Product structure 1 for the illustration problem

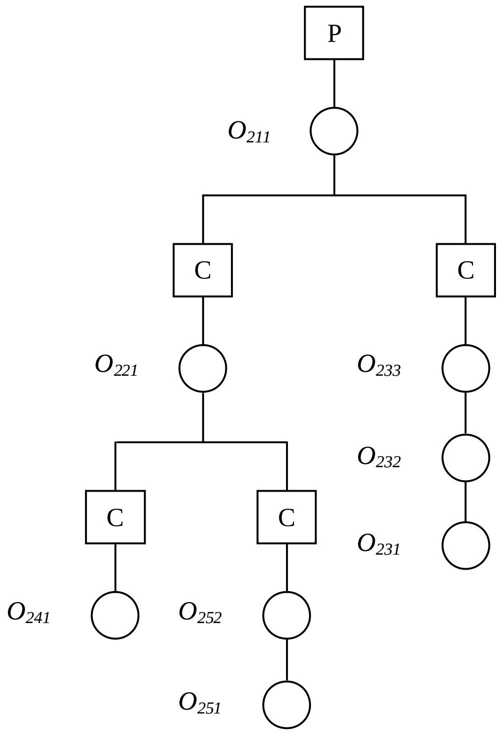

Figure 4.3 Product structure 2 for the illustration problem

The chromosome encoding scheme used is permutation encoding wherein each gene (an integer) in the string represents a particular processing or assembly operation from the set of all operations of all products and the position of the operation in the string indicates the priority of loading the operation on the machine while generating a feasible schedule using Giffler and Thompson schedule generation procedure (Giffler and Thompson, 1960, Girish and Jawahar, 2009). Therefore, the number of genes in the chromosome is equal to the total number of processing as well as assembling operations performed in each problem instances. One of the chromosomes generated for the initial population of the illustration problem is shown in Table 4.2.

Table 4.2 Information of a chromosome

	Chromosome c													
Gene No.	$g1$	$g2$	$g3$	$g4$	$g5$	$g6$	$g7$	$g8$	$g9$	$g10$	$g11$	$g12$	$g13$	$g14$
Value	O_{111}	O_{221}	O_{211}	O_{231}	O_{122}	O_{241}	O_{121}	O_{232}	O_{142}	O_{141}	O_{252}	O_{131}	O_{251}	O_{233}
Machine	$M1$			$M2$			$M3$			$M4$		$M5$		

Fitness evaluation: The fitness value of each chromosome is evaluated with a fitness function. The weighted sum of makespan and total tardiness corresponding to the chromosome '*c*' becomes the objective function or fitness value ($f(c)$) of it. The fitness value is modified with negative exponential function as $mf(c)=e^{-f(c)}$ in order to suit the probability of survival, where x is a scaling parameter (Girish and Jawahar, 2009). The best solution in the current population and the global best are sorted and stored separately.

New population generation: A new population is generated from the current population based on the concept of probability of survival. Roulette wheel selection method (Girish and Jawahar, 2009) has been adopted for the selection of chromosomes for new population. The chromosomes for crossover are selected based on the probability of crossover (P_c).

Due to the nature of the problem under consideration, there may be a chance of huge differences in the individual product tardiness values of the chromosomes even though they have equal or closer total tardiness values. When a crossover operation is performed among chromosomes with wider deviations in the individual product to product tardiness values, the resulting offsprings were found to be very poor solutions relatively to their parents in terms of the tardiness objective. Hence, to overcome this problem, a new pairing mechanism is developed to pair chromosomes for the crossover operation.

The procedure is as follows:

1. A pairing value matrix is calculated between all chromosomes based on their individual product tardiness values and is given below:
   Paring value between a chromosome *k* and a chromosome *l*,

   $$PV_{kl} = \left[\sum_{p=1}^{N} \left|T_{pk} - T_{pl}\right|^{a}\right] \tag{4.1}$$

   where T_{pk} and T_{pl} are the tardiness of chromosomes *k* and *l* for product *p*, *N* is the number of products and *a* is a scaling constant.
2. Parents are first selected for crossover based on the probability of crossover (P_c).
3. Corresponding to each parent selected in the step 2, the second parent is selected based on the roulette wheel selection procedure wherein the paring value between the first parent with other chromosomes is considered.
4. The selected pair of chromosomes then undergoes crossover operation.

Precedence preserving order based crossover (POX) (Lee et al., 1998, Pezzella et al., 2008) is the crossover operator used for schedule generation strings which maintains the precedence relationship among the operations in the chromosome after crossover. This operator generates two children starting from two parents. First, it randomly selects an operation from the first parent, copies in the first child with all its preceding and succeeding operations at the same place, then complete this new individual with the remaining operations, in the same order as they appear in the second parent. The symmetric process is repeated for the second parent and the second child. This operator preserves the sequencing constraints. The working of POX is shown in Table 4.3. The operation O_{221} is randomly selected from parent 1 for the first child and the corresponding set of preceding and succeeding operations is {O_{211}, O_{241}, O_{251}, O_{252}}. The operation O_{111} is randomly selected from parent 2 for the second child and the corresponding set of preceding and succeeding operations is {O_{121}, O_{122}, O_{131}, O_{141}, O_{142}}.

Table 4.3 Working of POX

Parent 1	Chromosome *c*													
Gene No.	*g1*	*g2*	*g3*	*g4*	*g5*	*g6*	*g7*	*g8*	*g9*	*g10*	*g11*	*g12*	*g13*	*g14*
Value	O_{111}	O_{221}	O_{211}	O_{231}	O_{122}	O_{241}	O_{121}	O_{232}	O_{142}	O_{141}	O_{252}	O_{131}	O_{251}	O_{233}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Parent 2	Chromosome *c*													
Gene No.	*g1*	*g2*	*g3*	*g4*	*g5*	*g6*	*g7*	*g8*	*g9*	*g10*	*g11*	*g12*	*g13*	*g14*
Value	O_{221}	O_{211}	O_{111}	O_{122}	O_{241}	O_{231}	O_{121}	O_{142}	O_{232}	O_{252}	O_{141}	O_{233}	O_{251}	O_{131}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Child 1	Chromosome *c*													
Gene No.	*g1*	*g2*	*g3*	*g4*	*g5*	*g6*	*g7*	*g8*	*g9*	*g10*	*g11*	*g12*	*g13*	*g14*
Value	O_{111}	O_{221}	O_{211}	O_{122}	O_{231}	O_{241}	O_{121}	O_{142}	O_{232}	O_{141}	O_{252}	O_{233}	O_{251}	O_{131}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Child 2	Chromosome *c*													
Gene No.	*g1*	*g2*	*g3*	*g4*	*g5*	*g6*	*g7*	*g8*	*g9*	*g10*	*g11*	*g12*	*g13*	*g14*
Value	O_{221}	O_{211}	O_{111}	O_{122}	O_{231}	O_{241}	O_{121}	O_{142}	O_{232}	O_{252}	O_{141}	O_{251}	O_{233}	O_{131}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Crossover is followed by mutation in which each gene of all the chromosomes is mutated with a probability of mutation (P_m). Precedence preserving swap operator is the mutation operator used for schedule generation strings which maintains the precedence relationship among the operations in the chromosome after mutation. This selects an operation from a single chromosome and swap with another randomly selected operation, taking care of the precedence constraints for those operations and its working is given in Table 4.4. The operation O_{122} is selected for mutation and it is swap with randomly selected operation O_{241} after checking its precedence constraints.

Table 4.4 Working of precedence preserving swap mutation

Parent	Chromosome *c*													
Gene No.	*g1*	*g2*	*g3*	*g4*	*g5*	*g6*	*g7*	*g8*	*g9*	*g10*	*g11*	*g12*	*g13*	*g14*
Value	O_{111}	O_{221}	O_{211}	O_{122}	O_{231}	O_{241}	O_{121}	O_{142}	O_{232}	O_{141}	O_{252}	O_{233}	O_{251}	O_{131}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Child	Chromosome *c*													
Gene No.	*g1*	*g2*	*g3*	*g4*	*g5*	*g6*	*g7*	*g8*	*g9*	*g10*	*g11*	*g12*	*g13*	*g14*
Value	O_{111}	O_{221}	O_{211}	O_{241}	O_{231}	O_{122}	O_{121}	O_{142}	O_{232}	O_{141}	O_{252}	O_{233}	O_{251}	O_{131}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Termination criteria: A specified number of iterations (N_{it}) are used to terminate the MOGA.

Output data: The MOGA prints a schedule corresponding to global best solution on satisfactory termination. Table 4.5 gives the final schedule for the illustration problem using MOGA.

Table 4.5 Final schedule of the illustration problem using MOGA

p	i	j	m	S_{pij}	C_{pij}	F_p	T_p	Makespan	Total tardiness
1	1	1	1	11	16	16	0	28	7
	2	1	3	0	8				
		2	2	8	11				
	3	1	5	0	4				
	4	1	4	0	2				
		2	3	8	9				
2	1	1	1	19	28	28	7		
	2	1	1	16	19				
	3	1	2	0	7				
		2	3	9	11				
		3	5	11	12				
	4	1	2	11	16				
	5	1	5	4	10				
		2	4	10	13				

4.3 MULTI-OBJECTIVE TABU SEARCH

4.3.1 Background

Tabu search (TS) is a metaheuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality. This technique was first suggested and applied by Glover (1989 and 1990) and since then has become increasingly used. A TS algorithm requires an initial solution to begin with and a move strategy to generate neighbourhood solutions. The basic idea of this technique is to search for the next candidate solution from among a carefully constructed neighborhood of the current solution by allowing for the possibility of the new solution being worse than the existing one. The algorithm maintains a list of prohibited moves, called tabu list, at each iteration of the search procedure. It is used to prevent the search from cycling between solutions. Generally, tabu list has a fixed size and when the list is full, the oldest element of the list is replaced by the new element. This makes tabu search a global optimizer rather than a local optimizer. In recent years, it has been applied to a wide range of combinatorial optimization problems. In this research, a multi-objective tabu search is proposed for assembly job shop scheduling problems with weighted approach.

4.3.2 Description of the proposed MOTS

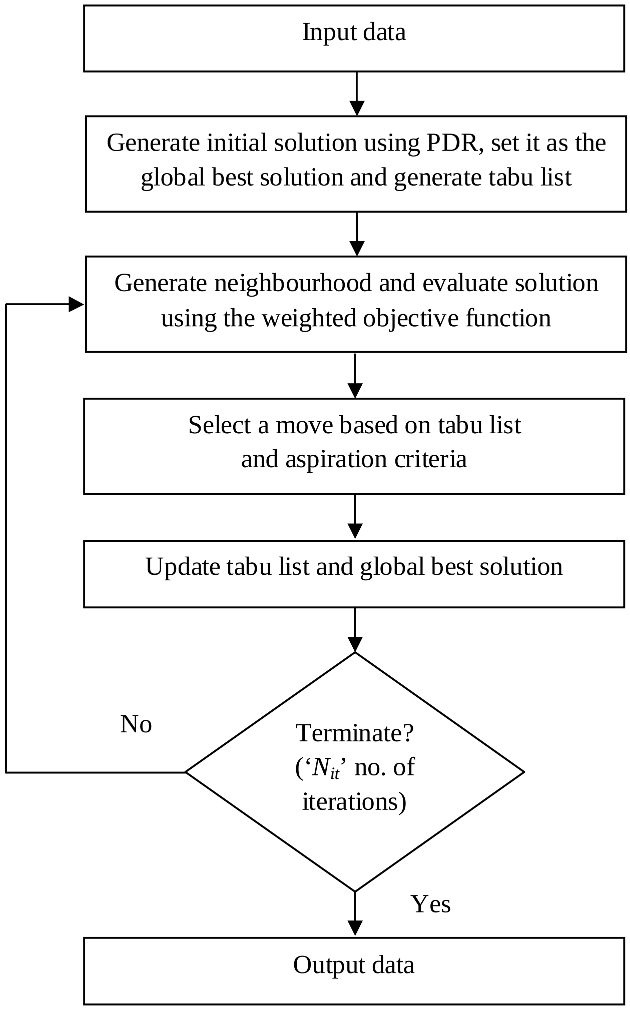

Figure 4.4 Procedure of the proposed MOTS

The procedure of the proposed MOTS is outlined in the flowchart given in Figure 4.4. The detailed description is as follows:

Input data: The input data as described in the proposed MOGA is given as the input.

Initial solution generation: It has been shown that the efficiency of the neighbourhood search methods depends closely on the quality of the initial solution. A feasible initial solution is obtained by selecting from among five due-date based priority dispatching rules as described in MOGA. This will help the proposed MOTS to reach a good quality final solution. Permutation encoding scheme is used for solution representation wherein each value (an integer) in the string represents a particular processing or assembly operation from the set of all operations of all products and the position of the operation in the string indicates the priority of loading the operation on the machine. Giffler and Thompson schedule generation procedure is used for generating a feasible schedule. (Giffler and Thompson, 1960, Girish and Jawahar, 2009). The solution string generated for illustration problem is shown in Table 4.6.

Table 4.6 Information of a solution string

	Solution string													
Position	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*p9*	*p10*	*p11*	*p12*	*p13*	*p14*
Value	O_{111}	O_{221}	O_{211}	O_{231}	O_{122}	O_{241}	O_{121}	O_{232}	O_{142}	O_{141}	O_{252}	O_{131}	O_{251}	O_{233}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Neighbourhood generation and evaluate weighted objective function value: A neighbourhood for a given solution is defined as the set of all permutations that can be created by a certain perturbation of the current solution. Insertion scheme (remove the operation from the current position and insert it into another position) is used for neighbourhood generation, taking care of the precedence constraints for that operation. The working of insertion scheme is given in Table 4.7. The objective function value of each neighbour is evaluated and the weighted sum of makespan and total tardiness corresponding to the neighbour '*n*' becomes the objective function of it.

Table 4.7 Working of insertion scheme

Parent	Solution string													
Position	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*p9*	*p10*	*p11*	*p12*	*p13*	*p14*
Value	O_{111}	O_{221}	O_{211}	O_{231}	O_{122}	O_{241}	O_{121}	O_{232}	O_{142}	O_{141}	O_{252}	O_{131}	O_{251}	O_{233}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

N 1	Solution string													
Position	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*p9*	*p10*	*p11*	*p12*	*p13*	*p14*
Value	O_{221}	O_{111}	O_{211}	O_{231}	O_{122}	O_{241}	O_{121}	O_{232}	O_{142}	O_{141}	O_{252}	O_{131}	O_{251}	O_{233}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Parent	Solution string													
Position	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*p9*	*p10*	*p11*	*p12*	*p13*	*p14*
Value	O_{111}	O_{221}	O_{211}	O_{231}	O_{122}	O_{241}	O_{121}	O_{232}	O_{142}	O_{141}	O_{252}	O_{131}	O_{251}	O_{233}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

N 2	Solution string													
Position	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*p9*	*p10*	*p11*	*p12*	*p13*	*p14*
Value	O_{221}	O_{211}	O_{111}	O_{231}	O_{122}	O_{241}	O_{121}	O_{232}	O_{142}	O_{141}	O_{252}	O_{131}	O_{251}	O_{233}
Machine	*M1*			*M2*			*M3*			*M4*		*M5*		

Move strategy: A move strategy is adopted to identify the most appropriate neighbour to which to move. The solutions in the current neighbourhood are examined in ascending order with respect to its objective function value and a neighbour S_0 is chosen if it is not tabu (i.e., not in the tabu list) or if an aspiration criterion is satisfied. The move function transforms S (current solution) into S_0 even if S_0 is not a better solution. The global best solution is replaced with current solution only if the current solution is better than the global best solution.

Tabu list: Tabu list of length l is introduced to prevent returning to a solution visited in the last l iterations, i.e, it stores all the forbidden moves. The tabu list size is fixed and once the list is full, the oldest element of the tabu list is removed as a new one is added. The working of tabu list is shown in the Table 4.8.

Table 4.8 Working of tabu list

Iteration 1	Tabu list													
Operation	O_{111}	O_{121}	O_{122}	O_{131}	O_{141}	O_{142}	O_{211}	O_{221}	O_{231}	O_{232}	O_{233}	O_{241}	O_{251}	O_{252}
Value	10													

Iteration 2	Tabu list													
Operation	O_{111}	O_{121}	O_{122}	O_{131}	O_{141}	O_{142}	O_{211}	O_{221}	O_{231}	O_{232}	O_{233}	O_{241}	O_{251}	O_{252}
Value	9				10									

Iteration 3	Tabu list													
Operation	O_{111}	O_{121}	O_{122}	O_{131}	O_{141}	O_{142}	O_{211}	O_{221}	O_{231}	O_{232}	O_{233}	O_{241}	O_{251}	O_{252}
Value	8				9					10				

Suppose that tabu list size is 10 and insertion of operation O_{111} makes the move in the first iteration. Hence it is not employed for the coming 10 iterations. The operation O_{141} makes move in the second iteration and it is not employed for the next 10 iterations, the operation O_{232} makes move in the third iteration and it is not employed for the coming 10 iterations and so on.

Aspiration criteria: An aspiration criterion could be such that it allows a tabu move when the neighbour has an objective function value better than the best objective encountered so far.

Termination criteria: A specified number of iterations (N_{it}) are used to terminate the MOTS.

Output data: The MOTS prints a schedule corresponding to global best solution on satisfactory termination. Table 4.9 gives the final schedule for the illustration problem using MOTS.

Table 4.9 Final schedule of the illustration problem using MOTS

p	i	j	M	S_{pij}	C_{pij}	F_p	T_p	Makespan	Total tardiness
1	1	1	1	11	16	16	0	28	7
	2	1	3	0	8				
		2	2	8	11				
	3	1	5	6	10				
	4	1	4	0	2				
		2	3	8	9				
2	1	1	1	19	28	28	7		
	2	1	1	16	19				
	3	1	2	0	7				
		2	3	9	11				
		3	5	11	12				
	4	1	2	11	16				
	5	1	5	0	6				
		2	4	6	9				

4.4 MULTI-OBJECTIVE HYBRID GENETIC ALGORITHM

4.4.1 Background

Hybrid algorithms exploit the good properties of different methods by applying them to problems they can efficiently solve. This is usually depends on the design quality of the hybrid algorithm. The genetic algorithms do very well in identifying the region in which the global optimum exists and they are not good to locate the exact local optimum in the region of convergence (De Jong, 2005). The difficulty of finding the best solution in the best found region accounts for the genetic algorithm operator's inability to make small moves in the neighbourhood of current solutions (Reeves, 1994). A combination of a genetic algorithm and a local search method can improve the exploiting ability of the search algorithm. In such a hybrid algorithm, applying a local search to the solutions that are guided by a genetic algorithm to the most promising region can accelerate convergence to the global optimum (El-Mihoub, 2006). Accordingly, the right combination of global exploration and local exploitation capabilities enable the hybrid genetic algorithm to produce high quality solutions. A multi-objective hybrid genetic algorithm in which the genetic algorithm coupled with a local search algorithm is investigated in this research for solving multi-objective assembly job shop scheduling problems with weighted approach.

4.4.2 Description of the proposed MOHA

The procedure of the proposed MOHA is outlined in the flowchart given in Figure 4.5. The input data requirements, initial population generation, fitness evaluation, new population generation, termination criteria and output data are same as described in MOGA. The procedure of the local search algorithm is shown in Figure 4.6 and the description is given below:

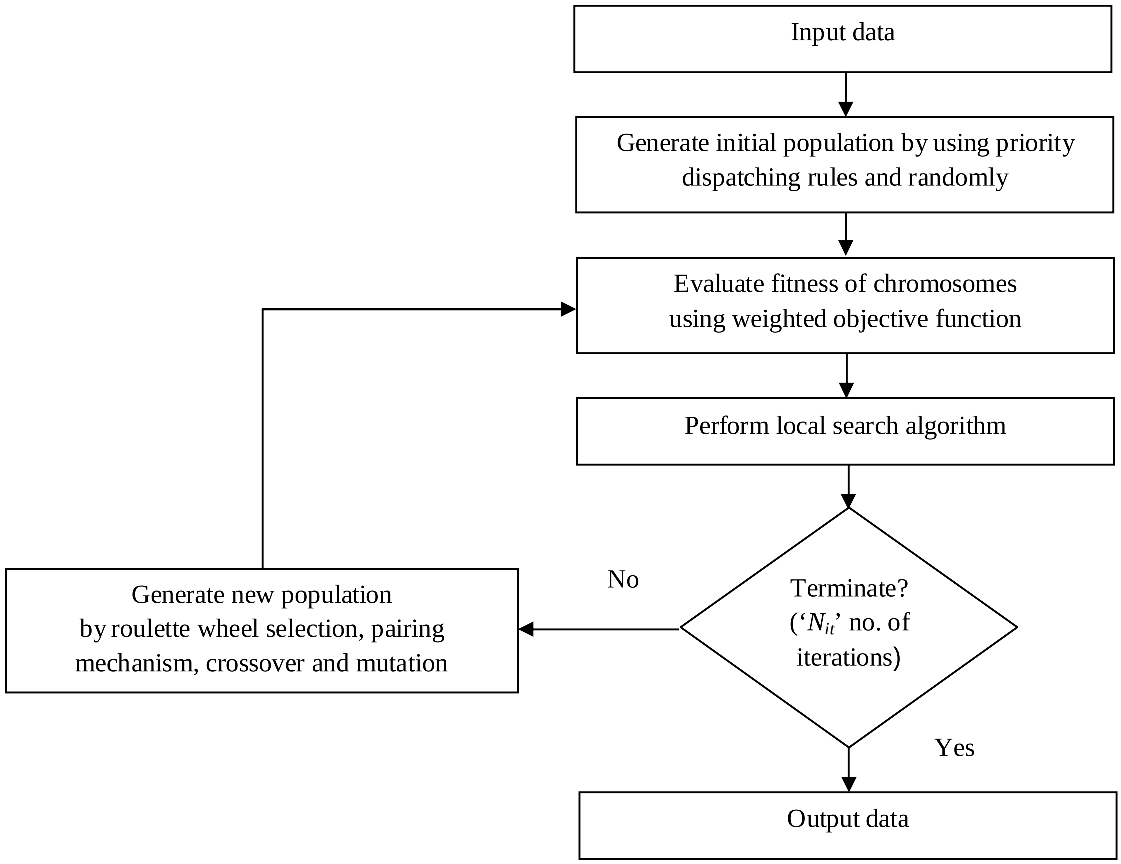

Figure 4.5 Procedure of the proposed MOHA

Local search algorithm: This is applied to all the chromosomes of the initial population and to the chromosomes modified by crossover and mutation in every iteration. A neighbourhood solution is obtained from its parent solution by means of a neighbourhood generation scheme. The insertion scheme in MOTS is also used here as the neighbourhood generation scheme. Every time a neighbourhood solution is found to be better than its parent solution, the neighbourhood solution becomes the parent solution for the subsequent iterations. The local search procedure is performed until no better solutions are found than its parent solution. The improved parent solution obtained from the local search procedure replaces the chromosome in the current generation and the corresponding fitness values are updated.

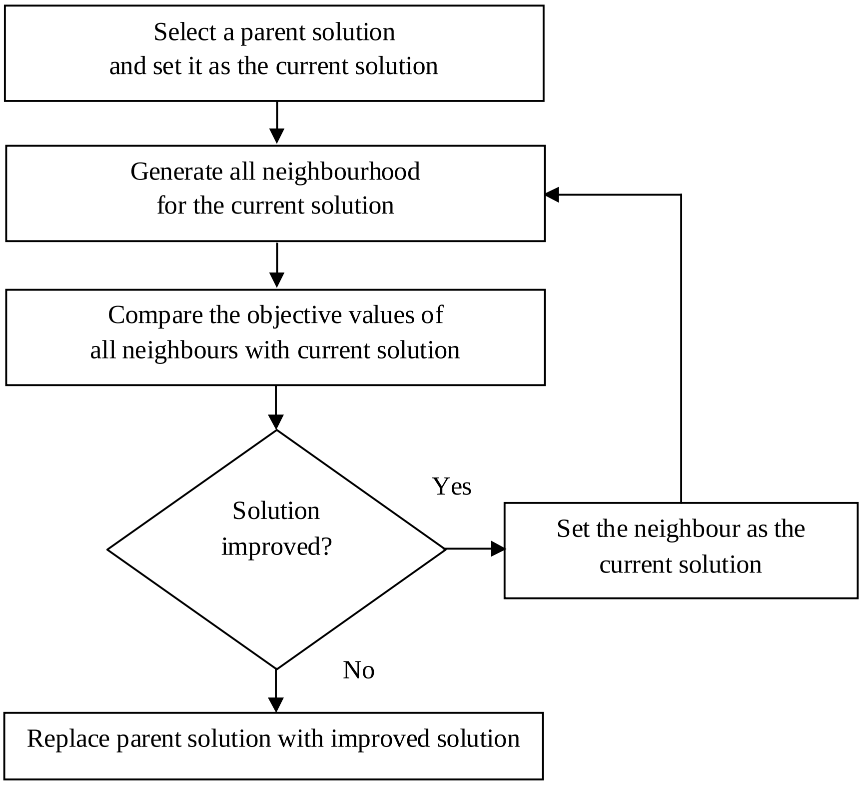

Figure 4.6 Procedure of the proposed local search algorithm for MOHA

Table 4.10 gives the final schedule for the illustration problem using MOHA. Since the illustration problem is a small size problem (2 products × 5 machines) with 14 operations, the final schedule is similar to one obtained with MOGA.

Table 4.10 Final schedule of the illustration problem using MOHA

<table>
<tr><th>p</th><th>i</th><th>j</th><th>m</th><th>S_{pij}</th><th>C_{pij}</th><th>F_p</th><th>T_p</th><th>Makespan</th><th>Total tardiness</th></tr>
<tr><td rowspan="6">1</td><td>1</td><td>1</td><td>1</td><td>11</td><td>16</td><td rowspan="6">16</td><td rowspan="6">0</td><td rowspan="14">28</td><td rowspan="14">7</td></tr>
<tr><td>2</td><td>1</td><td>3</td><td>0</td><td>8</td></tr>
<tr><td></td><td>2</td><td>2</td><td>8</td><td>11</td></tr>
<tr><td>3</td><td>1</td><td>5</td><td>0</td><td>4</td></tr>
<tr><td>4</td><td>1</td><td>4</td><td>0</td><td>2</td></tr>
<tr><td></td><td>2</td><td>3</td><td>8</td><td>9</td></tr>
<tr><td rowspan="8">2</td><td>1</td><td>1</td><td>1</td><td>19</td><td>28</td><td rowspan="8">28</td><td rowspan="8">7</td></tr>
<tr><td>2</td><td>1</td><td>1</td><td>16</td><td>19</td></tr>
<tr><td>3</td><td>1</td><td>2</td><td>0</td><td>7</td></tr>
<tr><td></td><td>2</td><td>3</td><td>9</td><td>11</td></tr>
<tr><td></td><td>3</td><td>5</td><td>11</td><td>12</td></tr>
<tr><td>4</td><td>1</td><td>2</td><td>11</td><td>16</td></tr>
<tr><td>5</td><td>1</td><td>5</td><td>4</td><td>10</td></tr>
<tr><td></td><td>2</td><td>4</td><td>10</td><td>13</td></tr>
</table>

4.5 SUMMARY

In this chapter three metaheuristic algorithms namely MOGA, MOTS and MOHA have been proposed for solving the MOAJSP with weighted approach. These algorithms are described in detail with the help of an illustration problem. The three algorithms are coded in C language and their performance evaluations are included in Chapter 6.

CHAPTER 5

ALGORITHMS WITH PARETO APPROACH

5.1 INTRODUCTION

Three metaheuristic algorithms based on genetic algorithm and tabu search with Pareto approach for solving multi-objective assembly job shop scheduling problem are discussed in this chapter. The background and description of the proposed algorithms namely Pareto archived genetic algorithm (PAGA), Pareto archived tabu search (PATS) and Pareto archived hybrid genetic algorithm (PAHA) are presented in the following sections.

5.2 PARETO ARCHIVED GENETIC ALGORITHM

5.2.1 Background

Evolutionary algorithm mimics nature's evolutionary principles to drive its search towards an optimum solution. An evolutionary algorithm can be used to capture multiple Pareto optimal solutions in one single simulation run for a multi-objective optimization problem (Deb et al., 2000a, 2000b). Through the years, a number of multi-objective evolutionary algorithms (MOEA) have been suggested by the researchers and the majority of them are based on genetic algorithms. Accordingly, most of the GA based evolutionary algorithms that attempt a controlled elitism in multi-objective optimization. Some of them are distance-based Pareto genetic algorithm (Osyczka and Kundu, 1995), strength Pareto evolutionary algorithm (Zitzler and Thiele, 1998), Pareto archived evolution strategy (Knowles and Corne, 2000), non-dominated sorting genetic algorithm - II (NSGA-II) (Deb et al., 2000a, 2000b) and strength Pareto evolutionary algorithm

2 (Zitzler et al., 2001). NSGA-II is commonly applied in various production scheduling environments. In this research, a Pareto archived genetic algorithm is developed based on the framework of NSGA-II (Deb et al., 2000a, 2000b) for solving multi-objective assembly job shop scheduling problems.

5.2.2 Description of the proposed PAGA

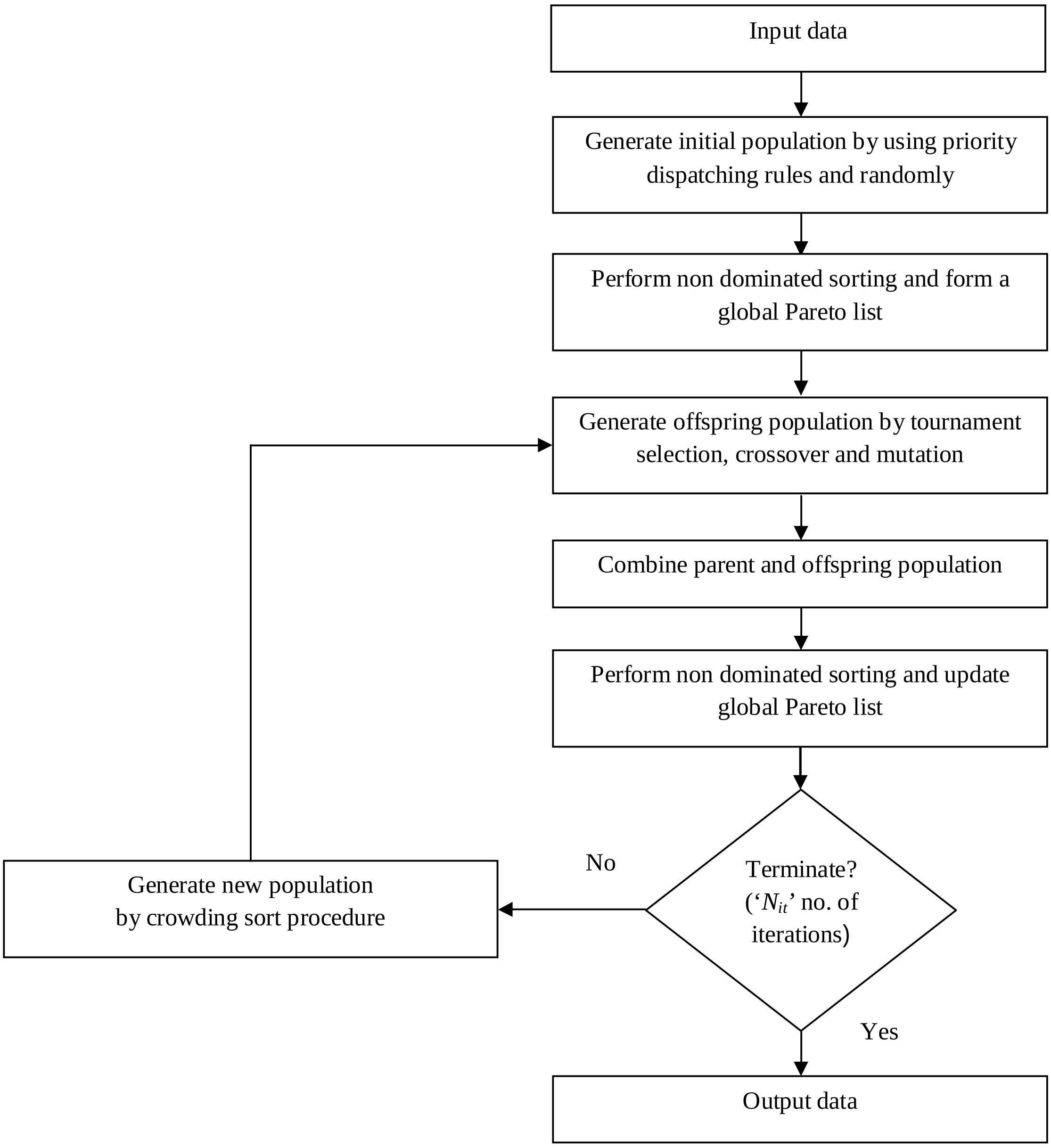

Figure 5.1 Procedure of the proposed PAGA

The procedure of the proposed PAGA is outlined in the flowchart shown in Figure 5.1. The detailed description is as follows:

Input data: The input data described in the proposed MOGA is given as the input.

Initial population generation: A set of chromosomes equal to the size of the population is generated as described in MOGA. Permutation encoding scheme is used for chromosome representation and Giffler and Thompson procedure is used for feasible schedule generation as similar to MOGA.

Non-dominated sorting: A fast non-dominated sorting approach (Deb et al., 2002) is used to classify the entire population into different non-domination levels. Two entities such as domination count (*DC*), the number of solutions which dominate the solution and a set of solutions that the solution dominates (*SD*) are calculated for each solution. All solutions with domination count as zero is included into the first non-dominated front. Subsequently, for each member in the set, *SD* (for each solution with domination count as zero) is visited and reduced its domination count by one. The solutions with domination count as zero is included in the second non-dominated front. This process continued until all fronts are identified. Each solution is assigned a fitness equal to its non-domination level (1 is the best level). Thus, minimization of fitness is assumed.

Offspring population generation: An offspring population is generated from the current population by using tournament selection with crowded comparison operator (Deb et al., 2000a, 2000b), crossover and mutation operators. Precedence preserving order based crossover (Lee et al., 1998, Pezzella et al., 2008) and Precedence preserving swap operator are used as the crossover and mutation operators respectively for schedule generation strings which maintains the precedence relationship among the operations in the chromosome after crossover and mutation.

Combined population creation: A combined population is created by combining parent and offspring population. This strategy gives an opportunity to ensure elitism.

New population generation: Crowding sort procedure (Deb et al., 2000a, 2000b) is used to generate a new population from the combined population. The combined population is sorted into different levels according to non-domination. The solutions from the best non-dominated front are definitely included in the new population, if the size of the front is smaller than the new population. The remaining members of the new population are chosen from subsequent non-dominated fronts in the order of their ranking. If more solutions are exist in the last front than the remaining slot in the new population, the solutions of the last front is arranged in descending order of their crowding distance values and choose the best solutions needed to fill the remaining slots in the new population.

Crowding distance estimation: The average distance of two solutions on either side of a particular solution along each of the objectives as called the crowding distance. This is usually calculated to get an estimate of the density of solutions surrounding a particular solution in the population. The crowding distance of a solution is taken as the perimeter of the cuboid formed by using the nearest neighbours as the vertices. The crowding distance assignment procedure (Deb et al., 2000a, 2000b) is given below:

- Sort the set according to each objective function value in ascending order of magnitude.
- Assign an infinite distance value to the boundary solutions for each objective function.
- Assign a distance value equal to the absolute normalized difference in the function values of two adjacent solutions for all other intermediate solutions for each objective function.

- Calculate the overall crowding distance value as the sum of individual distance values corresponding to each objective.

Crowded comparison operator: The crowded comparison operator (Deb et al., 2000a, 2000b) compares two solutions and returns the winner of the tournament. It assumes that every solution in the population has two attributes: non-domination rank and crowding distance. This is defined as follows:

A solution i is preferred over solution j if any one of the following conditions are true.

1. If the solution i has a better (lower) rank than j.
2. If they have the same rank, but the solution i has a better crowding distance than solution j.

Termination criteria: A specified number of iterations (N_{it}) are used to terminate the PAGA.

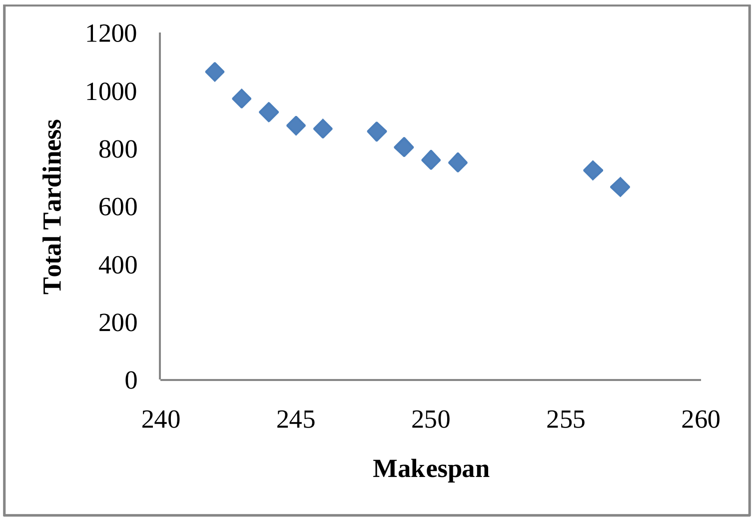

Figure 5.2 Pareto solutions obtained with PAGA for a 7 × 10 problem

Output data: The PAGA prints a set of solutions corresponding to the global Pareto front on satisfactory termination. The Pareto solutions obtained for a 7 products × 10 machines problem with PAGA is shown in Figure 5.2. The data regarding the problem is given in the appendix (Table A15).

5.3 PARETO ARCHIVED TABU SEARCH

5.3.1 Background

Tabu search, a neighbourhood based metaheuristic technique can be directly applied to any kind of optimization problem (Glover et al., 2007). The idea of applying tabu search to multiple-objective optimization comes from its solution structure, in working with more than one solution (neighbourhood solutions) at a time (Baykasoglu et al., 2004). The focus and emphasis of tabu search have only a few number of implications for the goal of designing Pareto based multi-objective optimization procedures (Kulturel-Konak et al., 2006). In this research, a multiple tabu search called Pareto archived tabu search is proposed for multi-objective assembly job shop scheduling problems.

5.3.2 Description of the proposed PATS

The procedure of the proposed PATS is outlined in the flowchart shown in Figure 5.3. The detailed description is as follows:

Input data: The input data described in the proposed MOGA is given as the input.

Initial population generation: A set of solutions equal to the size of the population is generated. The first five solutions in the initial population are generated from the due-date based priority dispatching rules as described in MOGA. The remaining solutions in the initial population are generated randomly. Permutation encoding scheme is used for solution representation as described in MOTS.

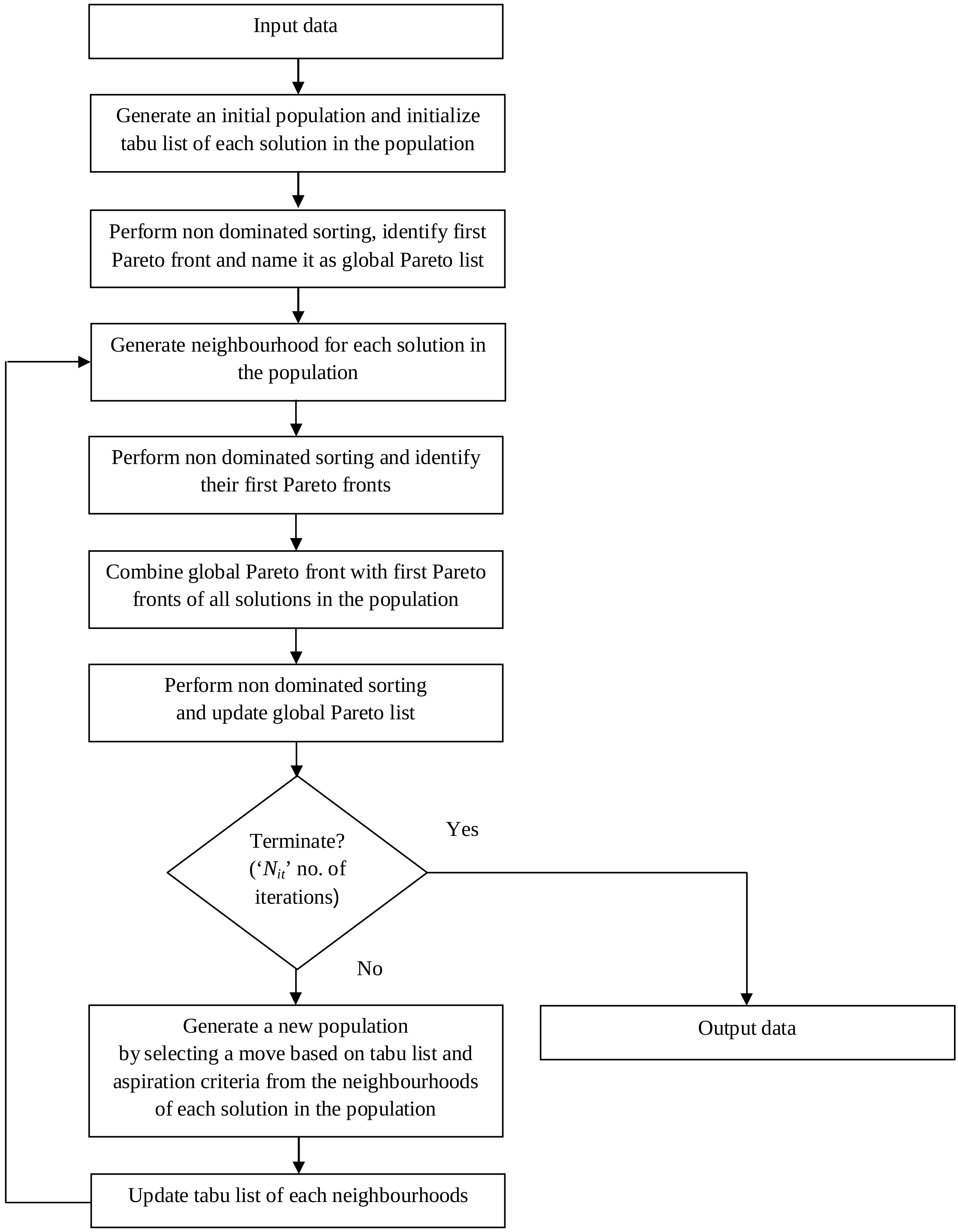

Figure 5.3 Procedure of the proposed PATS

Non-dominated sorting: A fast non-dominated sorting approach (Deb et al., 2002) is used to classify the entire population into different non-domination levels as described in PAGA.

Neighbourhood generation: Separate neighbourhoods are generated for each solution in the population and neighbourhood generation procedure is same as explained in MOTS.

Combined population: A combined population is created by combining global Pareto front and the first Pareto fronts of all neighbourhoods.

New population generation: A new population is generated by including a solution from neighbourhoods of each solution in the parent population. The solution selection is based on the individual tabu list and aspiration criteria for each neighbourhood.

Move strategy, tabu list and aspiration criteria: The move strategy and tabu list are as same as described in MOTS. Separate tabu lists are maintained for all neighbourhoods. An aspiration criterion could be such that it allows a tabu move when the neighbour dominates with solutions in the global Pareto front.

Termination criteria: A specified number of iterations (N_{it}) are used to terminate the PATS.

Output data: The PATS prints a set of solutions corresponding to the global Pareto front on satisfactory termination. The Pareto solutions obtained for a 7 products × 10 machines problem with PATS is shown in Figure 5.4.

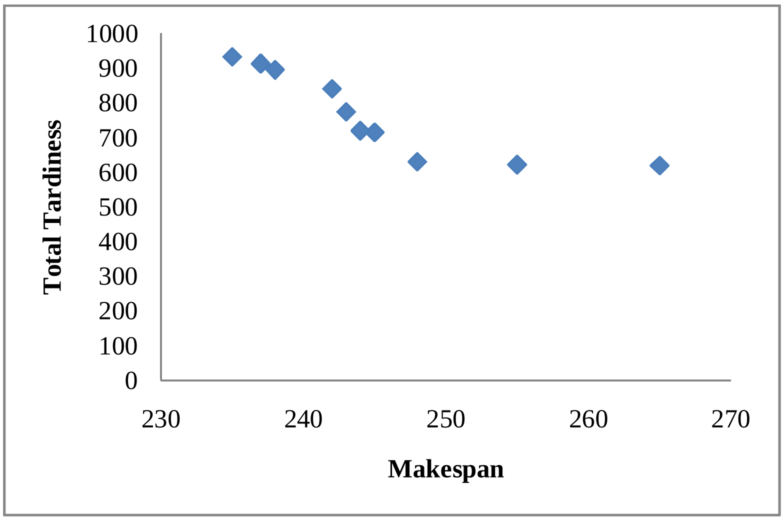

Figure 5.4 Pareto solutions obtained with PATS for a 7 × 10 problem

5.4 PARETO ARCHIVED HYBRID GENETIC ALGORITHM

5.4.1 Background

Hybrid algorithms in evolutionary optimization seem to be an emerging field nowadays due to its enhanced search capacity. Many researchers have focused their attention in designing hybrid evolutionary algorithms. Incorporating a local search method within a genetic algorithm can easily produce solutions with high accuracy (El-Mihoub, 2006). A Pareto archived hybrid genetic algorithm in which a genetic algorithm based on NSGA-II coupled with a local search algorithm is developed in this research for multi-objective assembly job shop scheduling problems.

5.4.2 Description of the proposed PAHA

The procedure of the proposed PAHA is outlined in the flowchart given in Figure 5.5. The input data requirements, initial population generation, non-dominated sorting, new population generation, offspring population generation, crowding distance estimation, crowded comparison operator, termination criteria and output data are same as described in PAGA. The pairing mechanism as used in MOGA has been used here for crossover operation. A combined population is also created by combining parent population, offspring population and local solutions list. The procedure of the local search algorithm is shown in Figure 5.6 and the description is given below:

Local search algorithm: A local search procedure is applied to the chromosomes in the offspring population. All possible neighbourhood solutions of the current solution generated by means of a neighbourhood generation scheme. The insertion scheme used in MOTS has been used here as the neighbourhood generation scheme. A Pareto front of the neighbourhood solutions is generated (local solutions list) and one solution from it is randomly chosen as the current solution for the subsequent iterations. The local solutions list is updated in every iteration. The local search procedure is performed until no better solutions are found than its current solution.

The Pareto solutions obtained for a 7 products × 10 machines problem with PAHA is shown in Figure 5.7.

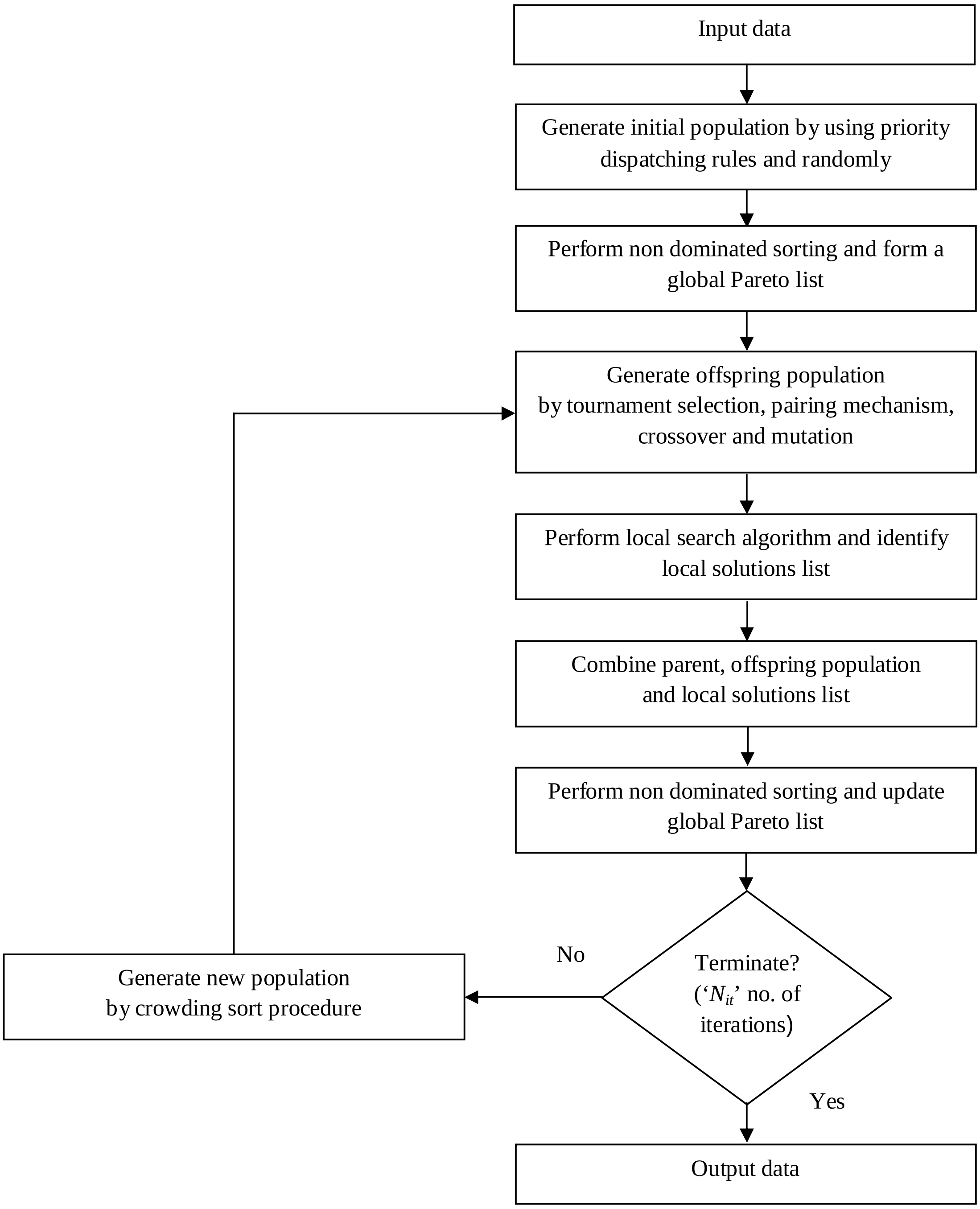

Figure 5.5 Procedure of the proposed PAHA

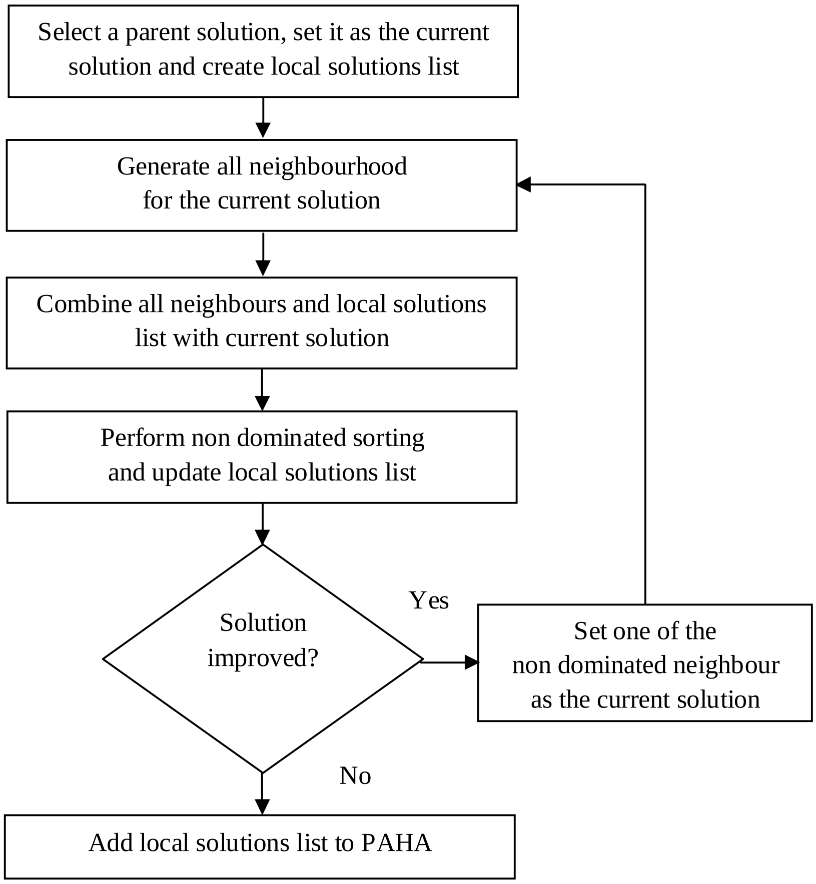

Figure 5.6 Procedure of the proposed local search algorithm for PAHA

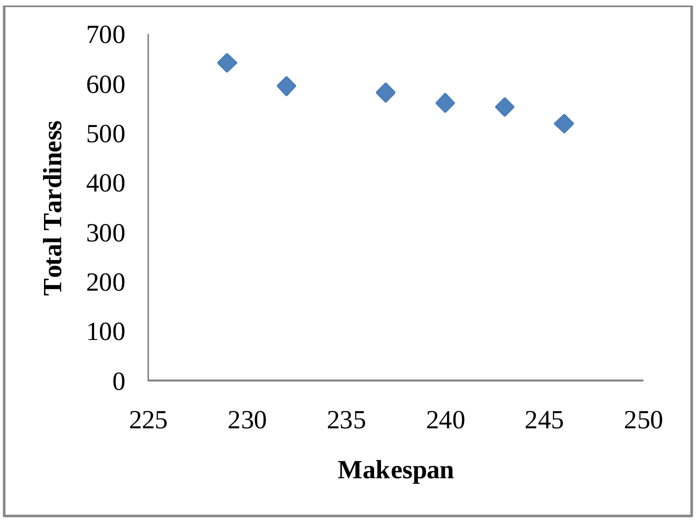

Figure 5.7 Pareto solutions obtained with PAHA for a 7 × 10 problem

5.5 SUMMARY

In this chapter three metaheuristic algorithms namely PAGA, PATS and PAHA have been proposed for solving the MOAJSP with Pareto approach. These algorithms are described in detail and Pareto optimal solutions obtained for a 7 products × 10 machines problem with the three algorithms is also given. The three algorithms are coded in C language and their performance evaluations are included in Chapter 6.

CHAPTER 6
RESULTS AND DISCUSSIONS

6.1 INTRODUCTION

This chapter presents the different numerical examples considered for the study. The results and discussion regarding the performance of the proposed algorithms for multi-objective assembly job shop scheduling problems is also presented.

6.2 NUMERICAL EXAMPLES

The performances of proposed algorithms are evaluated for various instances of multi-objective assembly job shop scheduling problem. Twenty eight different problem instances based on the literature are considered and the problem sizes vary from 3 products × 10 machines to 10 products × 15 machines. Three types of product structures namely flat product structure, tall product structure and complex product structures are included in each problem instance (Philipoom et al., 1991). The flat product structures have one level of assembly and between three to twelve components per assembly. The tall product structures have two to six levels of assembly and two components per assembly. The complex structures have two to three levels of assembly and two to four components per assembly. The examples of the three product structures are given in Figures 3.1 to 3.3 (Chapter 3). The processing and assembly times for operations at various levels are uniformly distributed in the range [1-15] for all the three types of product structures. The due-date of a product is calculated by multiplying the due-date allowance factor with sum of processing and assembling times on the critical path (Adam et al., 1993). The due date allowance factor is set at 1, 1.5 and 2, representing tight, medium and loose due-date setting, respectively. The list of different problem instances is given in Table 6.1.

Table 6.1 List of problem instances

Problem name	Problem size (N×M)	Number of components	Number of operations
P1	3×10	23	62
P2	3×10	28	73
P3	3×10	32	82
P4	3×10	31	85
P5	3×10	38	91
P6	3×10	38	98
P7	5×10	40	105
P8	5×10	41	110
P9	5×10	43	118
P10	5×10	43	122
P11	5×10	64	145
P12	5×10	63	156
P13	7×10	77	191
P14	7×10	79	207
P15	7×10	85	219
P16	7×10	86	225
P17	5×15	38	97
P18	5×15	41	109
P19	5×15	49	115
P20	5×15	44	116
P21	5×15	52	129
P22	5×15	57	144
P23	7×15	75	194
P24	7×15	75	199
P25	7×15	90	216
P26	7×15	84	225
P27	10×15	103	262
P28	10×15	98	268

The data pertaining to the above problem instances are given in Appendix.

The results of the proposed algorithms are evolved with the programs coded in C language and run in Visual C++ environment with an Intel Core i5 3.30 GHz Processor, 4GB RAM and CUDA (compute unified device architecture) enabled GTX 550 Ti Graphics Card. The local search part of the MOHA and PAHA algorithms were run using CUDA kernels on the Graphical Processing Units (GPU) due to its highly parallel nature.

6.3 ALGORITHMS WITH WEIGHTED APPROACH FOR MOAJSP

The performances of the three algorithms with weighted approach for MOAJSP are tested with the above problem instances. The parameters of the three proposed algorithms are obtained by fine tuning through trials. The details of parameters are given in Table 6.2.

Table 6.2 Parameter settings of MOGA, MOTS and MOHA

MOGA	
Population size	100
Crossover probability	30%
Mutation probability	5%
Factor *x*	0.05
Factor *a*	2
Number of iterations	N_{it}
MOTS	
Tabu size	30
Number of iterations	N_{it}
MOHA	
Population size	100
Crossover probability	30%
Mutation probability	5%
Factor *x*	0.05
Number of iterations	100
Factor *a*	2

Number of iterations (N_{it}) in MOGA and MOTS are calculated according to the number of solutions generated in MOHA for 100 iterations.

The performances of MOGA, MOTS and MOHA are evaluated with single objective criterion (makespan and total tardiness separately) and multi-objective criterion (weighted objective). In the present work, three sets of weights value (w_1 and w_2) have been considered ((0.3, 0.7), (0.5, 0.5) and (0.7, 0.3)) in the objective function.

The performances of the proposed algorithms are compared by using percentage reduction in solution (PRS) which is computed as follows

$$\text{PRS} = \left\{\frac{(\text{Solution by PDR} - \text{Solution by the algorithm})}{\text{Solution by the algorithm}}\right\} \times 100 \tag{6.1}$$

Solution by PDR is the best result obtained from the priority dispatching rules (PDR) namely operation due date (ODD), job due date (JDD), total work content remaining (TWKR) (Reeja and Rajendran, 2000b), shortest processing time (SPT) and first-in-first-out (FIFO) (Thiagarajan and Rajendran, 2003). The comparison of results for the proposed algorithms with makespan criterion is given in Table 6.3 and PRS is shown in Figure 6.1.

Table 6.3 Comparison of results for makespan criterion

Problem name	PDR	MOGA		MOTS		MOHA	
	Makespan	Makespan	PRS	Makespan	PRS	Makespan	PRS
P1	89	77	15.58	78	14.10	77	15.58
P2	90	90	0.00	90	0.00	90	0.00
P3	110	100	10.00	102	7.84	100	10.00
P4	112	110	1.82	110	1.82	110	1.82
P5	149	135	10.37	136	9.56	135	10.37
P6	130	125	4.00	125	4.00	125	4.00
P7	158	118	33.90	121	30.58	118	33.90
P8	134	129	3.88	129	3.88	129	3.88

Table 6.3 (Continued)

Problem name	PDR	MOGA		MOTS		MOHA	
	Makespan	Makespan	PRS	Makespan	PRS	Makespan	PRS
P9	159	134	18.66	138	15.22	134	18.66
P10	169	155	9.03	155	9.03	155	9.03
P11	187	172	8.72	180	3.89	171	9.36
P12	188	168	11.90	185	1.62	165	13.94
P13	253	232	9.05	239	5.86	227	11.45
P14	230	224	2.68	222	3.60	221	4.07
P15	255	230	10.87	232	9.91	228	11.84
P16	258	241	7.05	246	4.88	241	7.05
P17	94	92	2.17	92	2.17	91	3.30
P18	109	104	4.81	104	4.81	104	4.81
P19	105	94	11.70	99	6.06	89	17.98
P20	113	98	15.31	108	4.63	97	16.49
P21	115	109	5.50	111	3.60	108	6.48
P22	140	124	12.90	124	12.90	121	15.70
P23	183	142	28.87	167	9.58	139	31.65
P24	194	171	13.45	171	13.45	171	13.45
P25	195	177	10.17	186	4.84	178	9.55
P26	203	199	2.01	199	2.01	199	2.01
P27	249	206	20.87	217	14.75	202	23.27
P28	252	219	15.07	233	8.15	215	17.21

The results in the Table 6.3 and Figure 6.1 reveal that,

- For smaller size problems similar results are obtained with the three proposed algorithms (P1 to P10, P17 & P18).
- As the problem size increases MOHA performs better when compared to other algorithms (P11 to P16 & P19 to P28).
- Among MOGA and MOTS, MOGA performs better in most of the problem instances.

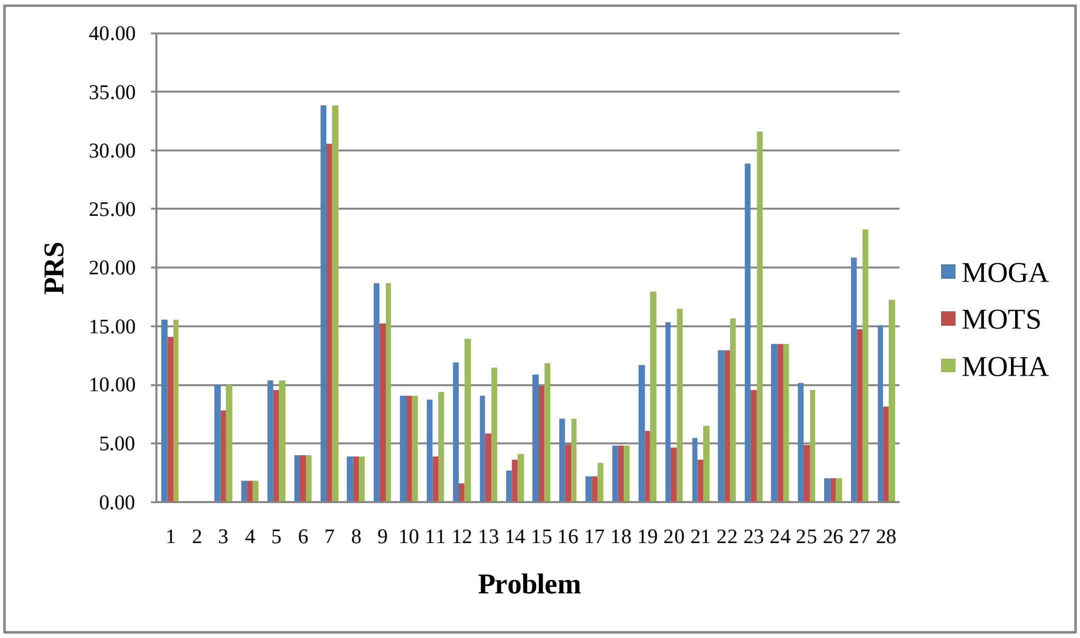

Figure 6.1 PRS for makespan criterion

The comparison of results for the proposed algorithms with total tardiness criterion is given in Table 6.4 and PRS is shown in Figure 6.2.

Table 6.4 Comparison of results for total tardiness criterion

Problem name	PDR	MOGA		MOTS		MOHA	
	Total tardiness	Total tardiness	PRS	Total tardiness	PRS	Total tardiness	PRS
P1	18	13	38.46	13	38.46	13	38.46
P2	73	43	69.77	43	69.77	37	97.30
P3	77	68	13.24	67	14.93	65	18.46
P4	111	74	50.00	78	42.31	73	52.05
P5	129	103	25.24	103	25.24	96	34.38
P6	128	89	43.82	119	7.56	82	56.10
P7	169	132	28.03	132	28.03	95	77.89
P8	226	151	49.67	177	27.68	115	96.52
P9	171	122	40.16	138	23.91	98	74.49
P10	269	187	43.85	194	38.66	149	80.54
P11	214	196	9.18	188	13.83	176	21.59
P12	197	163	20.86	153	28.76	135	45.93
P13	656	462	41.99	554	18.41	430	52.56

Table 6.4 (Continued)

Problem name	PDR	MOGA		MOTS		MOHA	
	Total tardiness	Total tardiness	PRS	Total tardiness	PRS	Total tardiness	PRS
P14	401	288	39.24	302	32.78	255	57.25
P15	696	574	21.25	620	12.26	515	35.15
P16	736	573	28.45	640	15.00	552	33.33
P17	94	57	64.91	59	59.32	48	95.83
P18	146	75	94.67	88	65.91	71	105.63
P19	42	17	147.06	19	121.05	10	320.00
P20	36	11	227.27	11	227.27	9	300.00
P21	127	99	28.28	103	23.30	86	47.67
P22	192	152	26.32	152	26.32	127	51.18
P23	376	219	71.69	246	52.85	175	114.86
P24	327	258	26.74	261	25.29	198	65.15
P25	317	284	11.62	300	5.67	260	21.92
P26	413	354	16.67	341	21.11	302	36.75
P27	670	517	29.59	582	15.12	436	53.67
P28	941	716	31.42	736	27.85	655	43.66

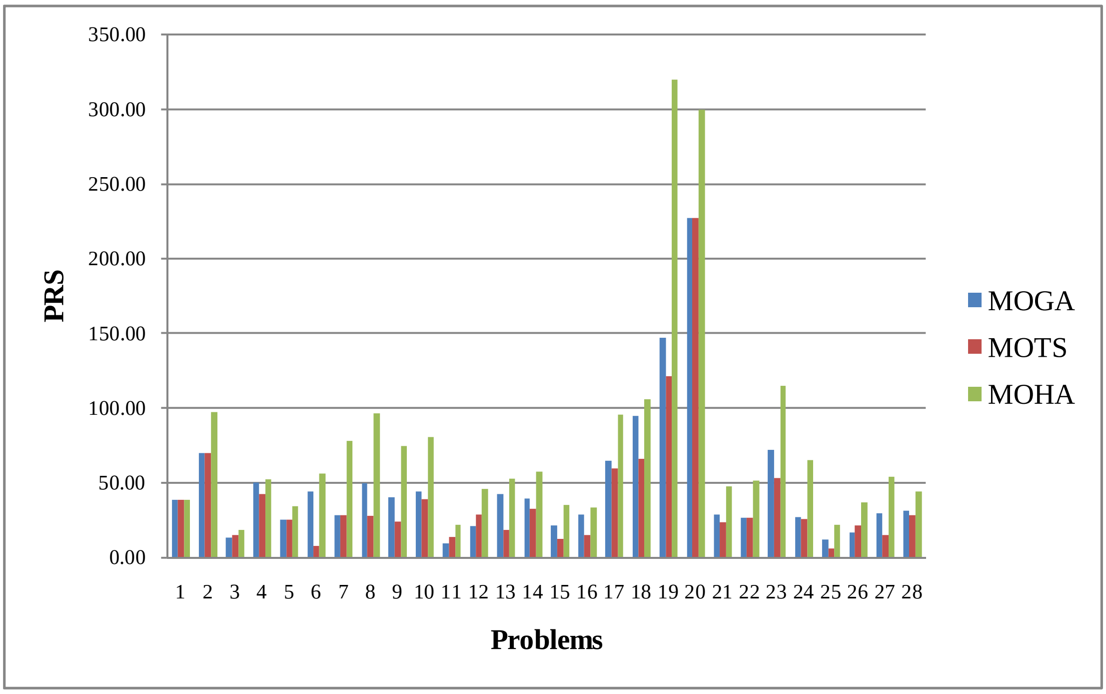

Figure 6.2 PRS for total tardiness criterion

The results in the Table 6.4 and Figure 6.2 reveal that,

- MOHA performs better than other two algorithms in almost all problem instances.
- Among MOGA and MOTS, MOGA performs better than MOTS in most of the problem instances.

The comparison of results for the algorithms with three sets of weight values are given in Tables 6.5 to 6.7 and PRS are shown in Figures 6.3 to 6.5.

Table 6.5 Comparison of results (W_1=0.3, W_2=0.7)

Problem name	PDR	MOGA		MOTS		MOHA	
	Solution	Solution	PRS	Solution	PRS	Solution	PRS
P1	42.90	34.90	22.92	34.90	22.92	34.90	22.92
P2	78.10	59.50	31.26	60.30	29.52	54.70	42.78
P3	91.10	84.00	8.45	81.70	11.51	80.30	13.45
P4	111.30	87.20	27.64	85.40	30.33	86.80	28.23
P5	135.60	114.40	18.53	114.40	18.53	109.60	23.72
P6	129.80	104.30	24.45	120.80	7.45	98.80	31.38
P7	171.40	134.10	27.82	136.90	25.20	109.40	56.67
P8	198.40	147.10	34.87	169.50	17.05	115.20	72.22
P9	168.90	132.80	27.18	143.10	18.03	118.20	42.89
P10	239.00	179.20	33.37	201.90	18.38	159.70	49.66
P11	205.90	192.80	6.79	196.80	4.62	181.10	13.69
P12	194.30	170.50	13.96	164.60	18.04	152.70	27.24
P13	537.50	395.80	35.80	465.50	15.47	365.00	47.26
P14	357.20	272.40	31.13	285.00	25.33	251.10	42.25
P15	570.00	489.00	16.56	495.60	15.01	432.50	31.79
P16	592.60	476.40	24.39	527.20	12.41	458.00	29.39
P17	97.60	69.00	41.45	71.60	36.31	62.70	55.66
P18	134.90	85.10	58.52	92.20	46.31	85.10	58.52
P19	65.70	45.80	43.45	47.40	38.61	37.90	73.35
P20	59.10	41.60	42.07	39.30	50.38	36.50	61.92
P21	129.70	106.70	21.56	105.70	22.71	94.30	37.54
P22	178.50	147.10	21.35	145.10	23.02	130.20	37.10
P23	326.80	207.80	57.27	226.80	44.09	178.10	83.49

Table 6.5 (Continued)

Problem name	PDR	MOGA		MOTS		MOHA	
	Solution	Solution	PRS	Solution	PRS	Solution	PRS
P24	296.40	234.50	26.40	223.40	32.68	191.50	54.78
P25	281.00	253.60	10.80	267.30	5.13	236.00	19.07
P26	352.40	309.70	13.79	301.70	16.80	278.20	26.67
P27	547.00	441.10	24.01	490.50	11.52	405.10	35.03
P28	746.60	582.00	28.28	590.80	26.37	538.90	38.54

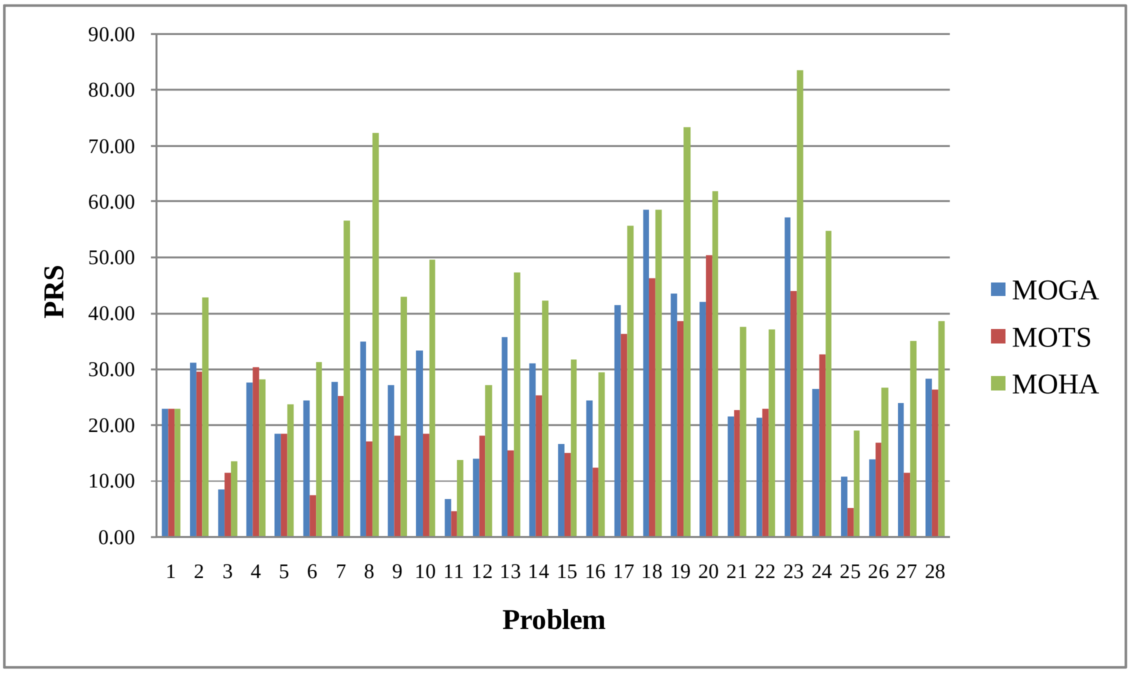

Figure 6.3 PRS (W_1=0.3, W_2=0.7)

Table 6.6 Comparison of results (W_1=0.5, W_2=0.5)

Problem name	PDR	MOGA		MOTS		MOHA	
	Solution	Solution	PRS	Solution	PRS	Solution	PRS
P1	59.50	49.50	20.20	50.00	19.00	49.50	20.20
P2	81.50	69.50	17.27	69.50	17.27	67.00	21.64
P3	100.50	93.50	7.49	91.50	9.84	90.50	11.05
P4	111.50	95.00	17.37	95.00	17.37	94.00	18.62
P5	140.00	121.50	15.23	122.00	14.75	118.00	18.64
P6	131.00	114.50	14.41	122.00	7.38	110.00	19.09
P7	171.00	130.00	31.54	133.50	28.09	114.50	49.34
P8	180.00	139.00	29.50	154.00	16.88	128.00	40.63
P9	167.50	140.00	19.64	146.00	14.73	127.50	31.37
P10	219.00	176.50	24.08	197.50	10.89	156.00	40.38
P11	200.50	189.00	6.08	192.00	4.43	179.00	12.01
P12	192.50	181.00	6.35	171.00	12.57	158.50	21.45
P13	458.50	352.00	30.26	406.50	12.79	320.50	43.06
P14	328.00	265.00	23.77	271.00	21.03	242.50	35.26
P15	486.00	418.00	16.27	430.50	12.89	377.50	28.74
P16	497.00	412.00	20.63	450.00	10.44	400.00	24.25
P17	100.00	76.50	30.72	80.00	25.00	72.50	37.93
P18	127.50	100.00	27.50	98.50	29.44	94.50	34.92
P19	81.50	62.50	30.40	68.00	19.85	56.50	44.25
P20	74.50	61.50	21.14	65.00	14.62	57.00	30.70
P21	131.50	109.00	20.64	113.50	15.86	101.00	30.20
P22	169.50	145.00	16.90	140.50	20.64	128.00	32.42
P23	292.50	200.50	45.89	213.00	37.32	170.00	72.06
P24	276.00	218.50	26.32	222.50	24.04	202.00	36.63
P25	257.00	237.00	8.44	245.50	4.68	220.00	16.82
P26	312.00	281.50	10.83	275.50	13.25	258.50	20.70
P27	465.00	382.50	21.57	408.50	13.83	352.00	32.10
P28	617.00	477.00	29.35	516.50	19.46	458.50	34.57

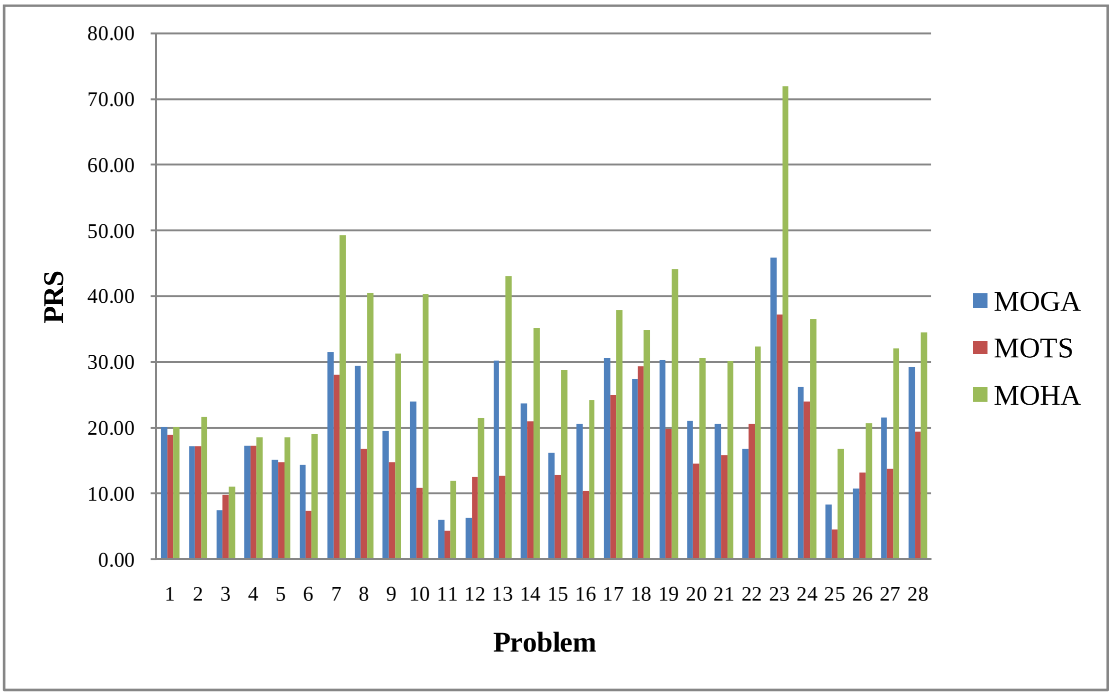

Figure 6.4 PRS (W_1=0.5, W_2=0.5)

Table 6.7 Comparison of results (W_1=0.7, W_2=0.3)

Problem name	PDR	MOGA		MOTS		MOHA	
	Solution	Solution	PRS	Solution	PRS	Solution	PRS
P1	73.50	64.10	14.66	64.40	14.13	64.10	14.66
P2	84.90	81.40	4.30	83.50	1.68	78.60	8.02
P3	109.90	94.90	15.81	94.60	16.17	94.60	16.17
P4	111.70	101.20	10.38	101.80	9.72	101.20	10.38
P5	144.40	128.50	12.37	128.80	12.11	123.60	16.83
P6	132.20	118.70	11.37	123.80	6.79	116.90	13.09
P7	170.60	125.00	36.48	132.50	28.75	120.40	41.69
P8	161.60	139.90	15.51	150.70	7.23	124.20	30.11
P9	166.10	144.10	15.27	149.60	11.03	128.40	29.36
P10	199.00	170.30	16.85	193.00	3.11	158.60	25.47
P11	195.10	184.20	5.92	187.20	4.22	173.90	12.19
P12	190.70	181.10	5.30	177.10	7.68	160.00	19.19
P13	379.50	308.00	23.21	347.50	9.21	289.40	31.13
P14	292.70	253.60	15.42	257.90	13.49	235.50	24.29
P15	402.00	357.70	12.38	373.50	7.63	315.70	27.34
P16	401.40	354.60	13.20	371.80	7.96	338.00	18.76

Table 6.7 (Continued)

Problem name	PDR	MOGA		MOTS		MOHA	
	Solution	Solution	PRS	Solution	PRS	Solution	PRS
P17	97.90	85.00	15.18	88.40	10.75	82.30	18.96
P18	120.10	101.80	17.98	104.80	14.60	100.10	19.98
P19	97.30	87.10	11.71	82.60	17.80	75.80	28.36
P20	89.90	82.10	9.50	79.70	12.80	79.40	13.22
P21	125.50	113.00	11.06	117.30	6.99	107.10	17.18
P22	160.50	141.30	13.59	135.80	18.19	129.00	24.42
P23	257.50	177.50	45.07	206.50	24.70	159.70	61.24
P24	249.90	200.60	24.58	205.00	21.90	190.10	31.46
P25	233.00	216.60	7.57	223.70	4.16	206.30	12.94
P26	271.60	251.20	8.12	254.00	6.93	235.00	15.57
P27	380.40	316.40	20.23	329.50	15.45	301.10	26.34
P28	487.40	382.50	27.42	416.40	17.05	366.60	32.95

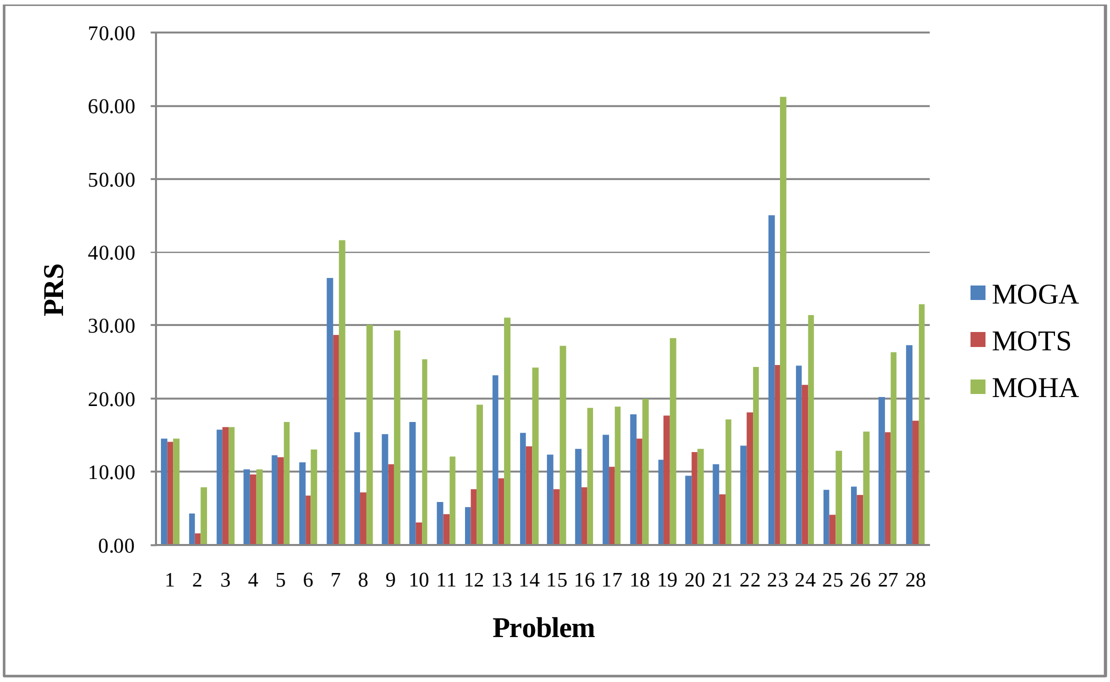

Figure 6.5 PRS (W_1=0.7, W_2=0.3)

The results reveal that,

- The proposed MOGA, MOTS and MOHA have a significant reduction in solution from the priority dispatching rules for all the problem instances under consideration (Tables 6.3 to 6.7, Figures 6.1 to 6.5).

- The proposed MOHA shows superiority for most of the MOAJSP instances with all the set of weight values considered when compared to MOGA and MOTS (Tables 6.5 to 6.7, Figures 6.3 to 6.5). This is due to the fact that GA is a global search method and is capable of rummaging through the solution space, however there is always a possibility that the algorithm may skip its local optimum solutions or the global optimum solution during its search. When local search technique is used, it takes the solution to its local optimum that in turn improves the convergence and performance of the algorithm. Hence problem under consideration being complex, GA-local search technique proves to be better than GA or local search alone.

- The performance of MOHA is also better than that of MOGA and MOTS in both single objective AJSP instances, makespan and total tardiness (Tables 6.3 & 6.4, Figures 6.1 & 6.2).

- The proposed MOGA is performed better than that of MOTS in most of the MOAJSP instances with all the set of weight values under consideration. In some instances the performance of MOTS is better than that of MOGA (Tables 6.5 to 6.7, Figures 6.3 to 6.5).

- In most of AJSP instances with makespan objective alone, the performance of MOGA and MOTS are almost similar to each other whereas MOGA outperformed MOTS in most of the AJSP instances with total tardiness criterion (Tables 6.3 & 6.4, Figures 6.1 & 6.2).

Hence it is concluded that the three proposed algorithms with weighted approach are feasible for solving MOAJSP instances. It is noted that the above algorithms are also suited for single objective AJSP instances.

6.4 ALGORITHMS WITH PARETO APPROACH FOR MOAJSP

The performances of the three algorithms with Pareto approach for MOAJSP are evaluated with the above test problems. The parameters of the three proposed algorithms are obtained by fine tuning through trials. The details of parameters are given in Table 6.8.

Table 6.8 Parameter settings of PAGA, PATS and PAHA

PAGA	
Population size	100
Crossover probability	100 %
Mutation probability	15 %
Number of iterations	N_{it}
PATS	
Population size	20
Tabu size	10
Number of iterations	N_{it}
PAHA	
Population size	100
Crossover probability	100 %
Mutation probability	15%
Factor a	2
Number of iterations	100

Number of iterations (N_{it}) in PAGA and PATS are calculated according to the number of solutions generated in PAHA for 100 iterations.

The Pareto optimal solutions obtained with the proposed PAGA, PATS and PAHA are given in Tables 6.9 to 6.11.

Table 6.9 Results obtained by PAGA

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P1	5	86	13
		84	23
		82	34
		81	54
		78	61
P2	3	97	52
		96	58
		90	68
P3	3	116	66
		101	83
		100	89
P4	3	120	90
		112	101
		110	109
P5	3	140	108
		139	110
		135	174
P6	3	146	122
		141	123
		125	124
P7	7	135	107
		132	114
		131	115
		129	152
		128	160
		121	177
		118	180

Table 6.9 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P8	3	137	130
		132	176
		129	202
P9	6	154	142
		141	158
		140	159
		137	161
		136	177
		135	179
P10	4	169	183
		165	223
		161	248
		155	313
P11	3	178	195
		177	331
		176	369
P12	8	191	174
		188	191
		185	205
		184	285
		181	287
		180	299
		178	304
		173	317
P13	9	276	696
		261	698
		256	705
		252	724
		250	735
		248	819
		245	855
		244	971
		243	1006

Table 6.9 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P14	6	255	401
		241	415
		232	429
		230	434
		229	565
		227	602
P15	11	257	668
		256	726
		251	753
		250	762
		249	806
		248	860
		246	870
		245	880
		244	927
		243	974
		242	1066
P16	8	258	736
		257	776
		256	835
		255	840
		250	866
		248	907
		247	954
		245	985
P17	4	98	58
		97	62
		94	74
		92	101
P18	4	118	71
		109	81
		105	115
		104	116

Table 6.9 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P19	11	115	14
		112	17
		110	19
		109	20
		107	26
		103	75
		101	83
		100	84
		95	117
		94	128
		91	145
P20	3	110	12
		107	46
		101	47
P21	5	122	106
		119	109
		115	113
		111	124
		110	165
P22	3	129	165
		128	188
		125	211
P23	6	166	307
		161	320
		160	330
		157	332
		154	345
		153	384
P24	8	213	305
		188	313
		186	330
		182	337
		180	351
		179	355
		172	384
		171	389

Table 6.9 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P25	8	197	317
		194	350
		193	351
		192	354
		188	356
		186	419
		185	466
		184	601
P26	6	216	390
		211	412
		210	416
		208	420
		203	490
		199	494
P27	11	260	670
		249	687
		244	738
		234	747
		225	758
		220	794
		218	834
		214	853
		213	921
		212	943
		211	1029
P28	10	257	879
		254	924
		253	941
		252	949
		231	957
		228	1072
		225	1098
		222	1132
		221	1146
		219	1171

Table 6.10 Results obtained by PATS

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P1	6	86	13
		84	23
		83	40
		82	51
		81	60
		77	61
P2	6	97	52
		96	58
		95	59
		93	62
		92	66
		90	68
P3	5	116	66
		115	91
		111	92
		104	102
		102	103
P4	3	112	93
		111	100
		110	102
P5	4	150	106
		147	107
		139	113
		135	175
P6	3	142	119
		140	122
		125	123
P7	7	136	138
		134	147
		132	152
		128	156
		127	225
		126	233
		125	248

Table 6.10 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P8	5	142	158
		140	179
		139	180
		130	182
		129	204
P9	7	155	143
		154	144
		151	169
		149	170
		138	177
		135	188
		134	311
P10	5	165	247
		162	267
		161	278
		156	306
		155	383
P11	3	178	184
		176	243
		174	250
P12	5	188	155
		187	159
		185	178
		183	290
		170	297
P13	10	252	541
		250	561
		248	573
		241	641
		240	645
		239	646
		238	656
		237	663
		233	738
		232	807

Table 6.10 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P14	9	238	382
		234	383
		230	399
		229	406
		228	447
		227	450
		226	530
		224	539
		223	830
P15	9	265	618
		255	621
		248	629
		245	714
		244	719
		243	774
		242	840
		237	913
		235	933
P16	9	258	677
		257	682
		256	686
		255	720
		250	730
		249	743
		245	794
		242	882
		241	1090
P17	5	98	58
		97	64
		94	80
		92	94
		91	109

Table 6.10 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P18	5	112	109
		110	117
		109	119
		105	124
		104	125
P19	8	117	19
		114	22
		109	24
		107	39
		104	52
		103	68
		97	74
		96	137
P20	5	110	7
		108	46
		106	52
		101	53
		98	69
P21	4	128	100
		123	111
		121	112
		109	165
P22	6	138	159
		132	161
		129	166
		128	193
		125	219
		124	255
P23	11	178	253
		175	254
		174	263
		171	271
		168	302
		164	352
		162	366
		159	370

Table 6.10 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
		158	376
		156	377
		154	405
P24	5	181	259
		180	275
		178	286
		177	288
		171	290
P25	4	189	304
		185	305
		184	309
		179	462
P26	4	210	371
		208	385
		202	419
		199	449
P27	4	227	645
		220	714
		215	861
		209	896
P28	10	258	766
		250	769
		248	797
		244	806
		240	883
		239	896
		223	964
		222	974
		220	978
		215	981

Table 6.11 Results obtained by PAHA

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P1	3	86	13
		84	23
		82	34
P2	5	97	37
		96	38
		95	42
		93	51
		90	64
P3	4	116	65
		115	78
		109	79
		100	80
P4	3	118	74
		112	76
		110	85
P5	3	139	99
		137	105
		135	173
P6	3	138	82
		137	85
		125	98
P7	4	129	102
		127	116
		123	124
		118	125
P8	5	139	132
		137	134
		135	142
		130	149
		129	156
P9	3	154	102
		135	113
		134	192

Table 6.11 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P10	4	165	169
		161	178
		156	213
		155	267
P11	3	178	182
		177	201
		171	210
P12	6	187	140
		179	146
		175	179
		173	223
		165	232
		163	259
P13	7	249	445
		237	452
		235	453
		234	455
		233	466
		232	619
		231	679
P14	8	236	256
		235	259
		231	265
		230	287
		224	458
		228	432
		226	447
		224	458
P15	6	246	519
		243	553
		240	561
		237	582
		232	595
		229	642

Table 6.11 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P16	4	255	557
		251	620
		247	664
		241	715
P17	6	98	49
		97	56
		94	61
		93	77
		92	92
		91	101
P18	4	118	76
		109	85
		108	90
		104	91
P19	6	105	10
		103	12
		102	15
		101	23
		96	47
		91	73
P20	3	111	6
		110	8
		101	21
P21	5	116	85
		115	89
		110	102
		108	115
		107	145
P22	4	127	130
		126	135
		125	138
		122	156

Table 6.11 (Continued)

Problem	No. of Pareto solutions	Pareto solutions	
		Makespan time	Total tardiness
P23	7	165	176
		162	189
		145	192
		143	234
		142	304
		141	309
		140	323
P24	3	180	207
		178	213
		177	256
P25	5	191	260
		189	267
		182	273
		177	316
		175	331
P26	3	212	307
		210	309
		199	333
P27	3	209	436
		206	615
		205	734
P28	6	248	685
		232	690
		230	695
		228	708
		221	756
		219	861

The performances of the proposed PAGA, PATS and PAHA are evaluated with a comparison metric called quality metric (Tavakkoli-Moghaddam et al., 2011a, 2011b). This metric is simply measured by putting together the non-dominated solutions found by algorithms and the ratios between non-dominated solutions are reported. The quality metric is given in Table 6.12 and Figure 6.6.

Metric C is also used to compare the approximate Pareto optimal set of the different algorithms (Zitzler and Thiele, 1999). $C(L, B)$ measures the fraction of members of B that are dominated by members of L.

$$C(L, B) = \frac{|\{b \in B : \exists h \in L, h > b\}|}{|B|} \tag{6.2}$$

where L and B are two sets of the decision vectors. $C(L, B) = 1$ implies that all solutions in B are dominated by solutions in L. $C(L, B) = 0$ means that none of solutions in B are dominated by the members of L.

Let L_1, L_2 and L_3 respectively indicate PAGA, PATS and PAHA. $C(L_i, L_j)$ indicates the fraction of all non-dominated members of archive of L_j that are dominated by all non-dominated members of archive obtained by L_i. $NDS(L_i, L_j)$ is the number of non dominated solutions finally obtained by L_j after the non dominated solutions of L_i have compared with those of L_j. Metric C is given in Tables 6.13 to 6.15.

Table 6.12 Quality metric

Problem name	PAGA	PATS	PAHA
P1	0	57.1429	42.8571
P2	0	0	100
P3	0	0	100
P4	0	0	100
P5	0	0	100
P6	0	0	100
P7	0	0	100
P8	25	0	75
P9	0	0	100
P10	0	0	100
P11	0	33.3333	66.6667
P12	0	0	100
P13	0	0	100
P14	0	0	100
P15	0	0	100

Table 6.12 (Continued)

Problem name	PAGA	PATS	PAHA
P16	0	0	100
P17	0	0	100
P18	50	0	50
P19	0	0	100
P20	0	50	50
P21	0	0	100
P22	0	0	100
P23	0	0	100
P24	0	25	75
P25	0	0	100
P26	0	0	100
P27	0	0	100
P28	0	14.2857	85.7143

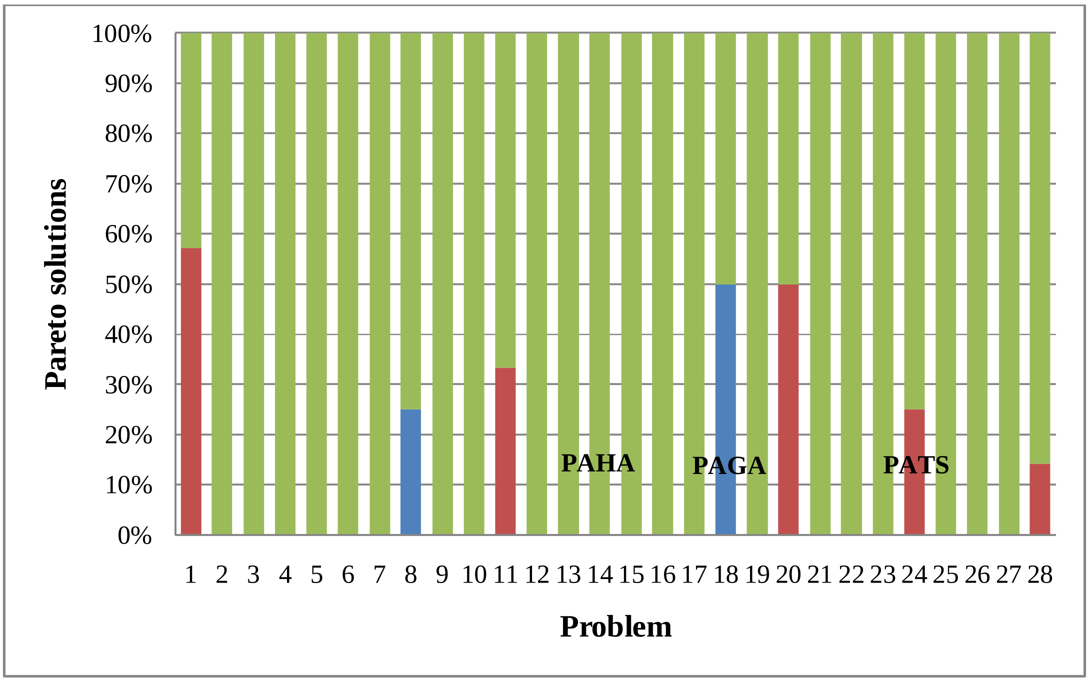

Figure 6.6 Quality metric

Table 6.13 Metric *C* obtained from PAGA and PATS

Problem name	$C(L_1, L_2)$	$NDS(L_1, L_2)$	$C(L_2, L_1)$	$NDS(L_2, L_1)$
P1	0.5000	3	0.2000	4
P2	0	3	0	3
P3	0.8000	1	0	3
P4	0	3	0.6667	1
P5	0.5000	2	0	3
P6	0	3	1	0
P7	0.8571	1	0.1429	6
P8	0.8000	1	0	3
P9	0.8571	1	0	6
P10	0.8000	1	0	4
P11	0	3	1	0
P12	0.2000	4	0.7500	2
P13	0	10	1	0
P14	0	9	1	0
P15	0	10	1	0
P16	0	9	1	0
P17	0.4000	3	0.2500	3
P18	1	0	0	4
P19	0.6250	3	0.2727	8
P20	0.6000	2	0.3333	2
P21	0.5000	2	0.2000	4
P22	0.5000	3	0	3
P23	0.5455	5	0	6
P24	0	5	1	0
P25	0	4	1	0
P26	0	4	1	0
P27	0.25	3	0.8182	2
P28	0	10	0.9	1

Table 6.14 Metric *C* obtained from PAGA and PAHA

Problem name	$C(L_1, L_3)$	$NDS(L_1, L_3)$	$C(L_3, L_1)$	$NDS(L_3, L_1)$
P1	0	3	0	5
P2	0	5	1	0
P3	0	4	1	0
P4	0	3	1	0
P5	0	3	1	0
P6	0	3	1	0

Table 6.14 (Continued)

Problem name	$C(L_1, L_3)$	$NDS(L_1, L_3)$	$C(L_3, L_1)$	$NDS(L_3, L_1)$
P7	0	4	1	0
P8	0.4000	3	0.6667	1
P9	0	3	1	0
P10	0	4	1	0
P11	0	3	1	0
P12	0	5	1	0
P13	0	3	1	0
P14	0	8	1	0
P15	0	6	1	0
P16	0	4	1	0
P17	0	6	1	0
P18	0.5000	2	0.5000	2
P19	0	6	1	0
P20	0	3	1	0
P21	0	5	1	0
P22	0	4	1	0
P23	0	7	1	0
P24	0	3	0.6667	3
P25	0	5	1	0
P26	0	3	1	0
P27	0	3	1	0
P28	0	6	1	0

Table 6.15 Metric *C* obtained from PATS and PAHA

Problem name	$C(L_2, L_3)$	$NDS(L_2, L_3)$	$C(L_3, L_2)$	$NDS(L_3, L_2)$
P1	0	3	0.3333	4
P2	0	5	1	0
P3	0	4	1	0
P4	0	3	1	0
P5	0	3	1	0
P6	0	3	1	0
P7	0	4	1	0
P8	0	5	1	0
P9	0	3	1	0
P10	0	4	1	0
P11	0.3333	2	0.6667	1

Table 6.15 (Continued)

Problem name	$C(L_2, L_3)$	$NDS(L_2, L_3)$	$C(L_3, L_2)$	$NDS(L_3, L_2)$
P12	0	5	1	0
P13	0	3	1	0
P14	0	8	0.8889	1
P15	0	6	1	0
P16	0	4	1	0
P17	0	6	1	0
P18	0	4	1	0
P19	0	6	1	0
P20	0.3333	2	0.6000	2
P21	0	5	1	0
P22	0	4	1	0
P23	0	7	1	0
P24	0	3	0.8000	1
P25	0	5	1	0
P26	0	3	1	0
P27	0	3	1	0
P28	0	6	0.9000	1

The results reveal that,

- The proposed PAHA is superior to PAGA and PATS in all test problem instances (Table 6.12 and Figure 6.6). Similar to the case of MOHA, PAHA uses a local search technique which improves solution in every iteration that leads to improved solutions and faster convergence. In addition to that, the pairing mechanism used for pairing of chromosomes for crossover operation in PAHA improves the solution quality by a larger extent.

- The proposed PAHA produces more number of non-dominated solutions when compared to PAGA and PATS (Tables 6.14 & 6.15).

- The proposed PATS produces more number of non-dominated solutions than PAGA in all larger size problem instances with few exceptions (P13 to P16 and P24 to P28) whereas the PAGA produces more non-dominated solutions than PATS in most of the small and medium size problem instances (Table 6.13).

Hence it is concluded that the proposed PAGA, PATS and PAHA are feasible and effective in generating Pareto optimal scheduling plans for MOAJSP instances, with PAHA outperforming the other two approaches for large size problem instances.

6.5 SUMMARY

The six different algorithms with two approaches viz., weighted approach and Pareto approach are evaluated for a set of test problems. The computational results reveal the efficiency of the proposed algorithms. MOHA and PAHA are found to be superior over others. The Pareto solutions obtained with PAHA are found to be closer to the results obtained by MOHA through weighted approach. The algorithms based on weighted approach are also useful for solving single objective AJSP instances. Concluding remarks and scope for future research are given in Chapter 7.

CHAPTER 7

CONCLUSION AND SCOPE FOR FUTURE RESEARCH

7.1 CONCLUSION

The scheduling of assembly job shop problem with multiple objectives is addressed in this thesis. An exhaustive literature survey is conducted and it reveals that the scheduling of assembly job shop with multiple objectives is seldom considered by the researchers though it has significant practical interest.

Two multi-objective solution approaches namely, weighted approach and Pareto approach are used for solving the multi-objective assembly job shop scheduling problem, which is a complex combinatorial optimization problem. Due to its complexity, two metaheuristic algorithms namely, genetic algorithm and tabu search are investigated in this thesis for scheduling assembly job shops with multiple objectives.

Three algorithms namely, multi-objective genetic algorithm, multi-objective tabu search and multi-objective hybrid genetic algorithm are developed to solve the problem with weighted approach. Similarly three algorithms namely, Pareto archived genetic algorithm, Pareto archived tabu search and Pareto archived hybrid genetic algorithm are developed for solving the problem with Pareto approach.

A new pairing mechanism is developed for pairing chromosomes for crossover operation and adopted in MOGA, MOHA and PAHA. This mechanism is effective for improving total tardiness criterion for the problems under consideration.

An exhaustive computational study is conducted to prove the efficiency of the proposed algorithms. The results show that the proposed MOGA, MOTS and MOHA are viable for scheduling assembly job shops with respect to the weights assigned to the objective functions. Among the three algorithms MOHA is more effective than others for solving MOAJSP instances. The computational results also proved that the proposed PAGA, PATS and PAHA are capable of producing a number of Pareto optimal scheduling plans. In terms of quality of solutions obtained, the proposed PAHA is superior to others.

The proposed algorithms provide schedules for all operations involved. The performance analysis reveals that the proposed algorithms are effective in minimizing makespan and total tardiness for assembly job shop scheduling problems.

7.2 SCOPE FOR FUTURE RESEARCH

There are a number of further works that can be considered in the future. Some of the future research directions are outlined as follows:

- The multi-objective assembly job shop scheduling problem is considered in this thesis assumes that each operation should be performed on a particular machine and each component has strictly ordered operation sequences. However, many real life environments allow routing flexibility for each operation and sequencing flexibility among operations of each component. This issue may be included in the future research.
- The problem environment considered in thesis assumes setup independence and assigns equal priorities to all components/products. Many industrial environments are setup dependent and the component/product may have different priority. The model would be extended by considering this issue.

- The proposed algorithms are based on genetic algorithm and tabu search. The application of other meta-heuristics to multi-objective assembly job shop scheduling can be considered in the future research.
- The problem environment considered in this thesis is based on the literature and can be extended for practical application in future.

REFERENCES

1. Adam, N. R., Bertrand J. W. M., and Surkis, J. (1987) Priority assignment procedures in multi-level assembly job shops, *IEE Transactions*, 19 (3), 317-328.

2. Adam, N. R., Bertrand, J. W. M., Morehead, D. C. and Surkis J. (1993) Due date assignment procedures with dynamically uploaded coefficients for multi level assembly job shops, *European Journal of Operational Research,* 68 (2), 212-227.

3. Baker, K. R. (1974) *Introduction to Sequencing and Scheduling,* Wiley, New York.

4. Baykasoglu, A., Ozbakir, L., and Sonmez, A. I. (2004) Using multiple objective tabu search and grammars to model and solve multi-objective flexible job shop scheduling problems, *Journal of Intelligent Manufacturing,* 15, 777-785

5. Brucker, P. (1995) *Scheduling Algorithms,* Springer-Verlag, Berlin-Heidelberg.

6. Chan, F. T. S., Wong, T. C., and Chan, L. Y. (2008a) Lot streaming for product assembly in job shop environment, *Robotics and Computer-Integrated Manufacturing,* 24, 321–331.

7. Chan F. T. S., Wong, T. C., and Chan, L. Y. (2008b) An evolutionary algorithm for assembly job shop with part sharing, *Computers & Industrial Engineering,* 1-11.

8. Chan, F. T. S., Wong, T. C., and Chan, L. Y. (2008c) The application of lot streaming to assembly job shop under resource constraints, *Proceedings of the 17th IFAC World Congress,* Seoul, Korea, 14852-14857.

9. Cheng, T. C. E. (1994) Optimal due date assessment in an assembly shop, *International Journal of Operations and Production Management,* 14 (2), 31-42.

10. Consoli, S. (2006) Combinatorial optimization and metaheuristics, *Operational Research Report: TB-01-06-Brunel Report,* School of Information Systems, Computing and Mathematics, Brunel University.

11. Conway, R. W., Maxwell, W. L., and Miller, L. M. (1967) *Theory of Scheduling,* Addison-Wesley Reading, MA.

12. Deb, K. (2001) *Multi-Objective Optimization using Evolutionary Algorithms,* John Wiley and Sons, West Sussex.

13. Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000a) A fast elitist non dominated sorting genetic algorithm for multi-objective optimization: NSGA II, *Kanpur Genetic Algorithms Laboratory (KanGAL) Report No. 200001,* 1-11.

14. Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000b) A fast and elitist non dominated sorting genetic algorithm for multi-objective optimization: NSGA II, *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI),* 849-858.

15. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation,* 6 (2), 182-197.

16. De Jong, K. (2005) Genetic algorithms: a 30 year perspective, *Perspectives on Adaptation in Natural and Artificial Systems,* Booker, L., Forrest, S., Mitchell, M., and Riolo, R. (Eds.), Oxford University Press.

17. Dimyati, T. T. (2007) Minimizing production flow time in a process and assembly job shop, *Proceedings of the International Seminar on Industrial Engineering and Management,* Jakarta, A68-A73.

18. Doctor, S. R., Cavalier, T. M., and Egbelu P. J. (1993) Scheduling for machining and assembly in a job-shop environment, *International Journal of Production Research,* 31 (6), 1275-1297.

19. El-Mihoub, T. A., Hopgood, A. A., Nolle, L., and Battersby, A. (2006) Hybrid genetic algorithms: a review, *Engineering Letters*, 13 (2), 124-137.

20. Fattahi, P. (2009) A hybrid multi objective algorithm for flexible job shop scheduling, *World Academy of Science, Engineering and Technology,* 50, 551-556.

21. Fattahi, P., Mehrabad, M. S., and Jolai, F. (2007) Mathematical modeling and heuristic approaches to flexible job shop scheduling problems, *Journal of Intelligent Manufacturing*, 18, 331-342.

22. Fry, T. D., Oliff, M. D., Minor, E. D., and Leong, G. K. (1989) Effects of product structure and sequencing rule on assembly shop performance, *International Journal of Production Research*, 27 (4), 671-686.

23. Giffler, B. and Thompson, G. L. (1960) Algorithms for solving production scheduling problems', *Operations Research 8*, 487–503.

24. Girish, B. S. (2009) Population based search heuristics for scheduling job shop associated with multiple routings, *Unpublished Ph.D. Thesis,* Anna University, Chennai.

25. Girish, B. S. and Jawahar, N. (2009) Scheduling job shop associated with multiple routings with genetic and ant colony heuristics, *International Journal of Production Research,* 47 (14), 3891–3917.

26. Glover, F. (1989) Tabu search – Part I, *ORSA Journal on Computing,* 1 (3), 190-206.

27. Glover, F. (1990) Tabu search – Part II, *ORSA Journal on Computing,* 2 (1), 4-32.

28. Glover, F., Laguna, M., and Marti, R. (2007) Principles of tabu search, *Handbook of Approximation Algorithms and Metaheuristics,* 23, 1-12.

29. Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning,* Addison-Wesley Reading, MA.

30. Gomes, M. C., Barbosa-Póvoa, A., and Augusto Q. Novais, A. Q. (2009) Scheduling of job shop, make-to-order industries with recirculation and assembly: discrete versus continuous time models, *Proceedings of Multidisciplinary International Conference on Scheduling: Theory and Applications*, Dublin, Ireland, 802-807.

31. Groover, M. P. (2003) *Automation, Production Systems and Computer Integrated Manufcaturing,* Prentice Hall of India, New Delhi.

32. Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T., and Chan, S. F. (2006) Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: a case study based on the apparel industry, *Computers & Industrial Engineering,* 50, 202–219.

33. Hariri, A. M. A., and Potts, C. N. (1997) A branch and bound algorithm for the two-stage assembly scheduling problem, *European Journal of Operational Research,* 103, 547-556.

34. Hicks, C. and Poncharoen, P. (2006) Dispatching rules for production scheduling in the capital goods industry, *International Journal of Production Economics*, 104, 154-163.

35. Ho, N. B. and Tay J. C. (2005) LEGA: an architecture for learning and evolving flexible job-shop schedules, *IEEE,* 1380-1387.

36. Ho, N. B., Tay, J. C., and Lai, E. M. K. (2007) An effective architecture for learning and evolving flexible job-shop schedules, *European Journal of Operational Research*, 179, 316–333.

37. Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems,* Ann Arbor, MI:MIT Press.

38. Hsu, T., Dupas, R., Jolly, D. and Goncalves, G. (2002) Evaluation of mutation heuristics for the solving of multiobjective flexible job shop by an evolutionary algorithm, *In: Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics,* 6–9.

39. Huang, G. Q. and Lu, H. (2009) A bilevel programming approach to assembly job shop scheduling, *Proceedings of IEEE Conference*, 182-187.

40. Hutchion, J. (1991) Current and future issues concerning FMS scheduling, *International Journal of Management Sciences*, 19 (6), 529-537.

41. Jang, Y., Kim, K., Jang, S., and Park, J. (2003) Flexible job shop scheduling with multi-level job structures, *JSME International Journal*, Series C, 46 (1), 33-38.

42. Jawahar, N., Aravindan, P., and Ponnambalam, S. G. (1998) A genetic algorithm for scheduling flexible manufacturing systems, *International Journal of Advanced Manufacturing Technology*, 14, 588-607.

43. Jia, Z. and Chen, H., and Tang J. (2007a) An improved particle swarm optimization for multi-objective flexible job-shop scheduling problem, *Proceedings of IEEE International Conference on Grey Systems and Intelligent Services*, Nanjing, China, 1587-1592.

44. Jia, Z. and Chen, H., and Tang J. (2007b) A new multi-objective fully-informed particle swarm algorithm for flexible job-shop scheduling problems, *Proceedings of International Conference on Computational Intelligence and Security Workshops*, 191-194.

45. Kacem, I., Hammadi, S., and Borne, P. (2002a) Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems, *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 32 (1), 1-13.

46. Kacem, I., Hammadi, S., and Borne, P. (2002b) Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic, *Mathematics and Computers in Simulation*, 60, 245–276.

47. Kachitvichyanukul, V. and Sitthitham, S. (2011) A two-stage genetic algorithm for multi-objective job shop scheduling problems, *Journal of Intelligent Manufacturing*, 22 (3), 355-365.

48. Kim J. U. and Kim Y. D. (1996) Simulated annealing and genetic algorithms for scheduling products with multi-level product structure, *Computers and Operations Research*, 23 (9), 857-868.

49. Knowles, J. D. and Corne, D. W. (2000) Approximating the non-dominated front using the Pareto archived evolution strategy, *Evolutionary Computation Journal*, 8 (2), 149-172.

50. Kolisch, R. (2001) *Make-to-order Assembly Management*, Springer, New York.

51. Kulturel-Konak, S., Smith, A. E., and Norman, B. A. (2006) Multi-objective tabu search using a multinomial probability mass function, *European Journal of Operational Research*, 169, 918–931.

52. Lagodimos, A. G., Mihiotis, A. N., and Kosmidis, V. C. (2004) Scheduling a multi-stage fabrication shop for efficient subsequent assembly operations, *International Journal of Production Economics*, 90, 345-359.

53. Lee, K. M., Yamakawa, T., and Lee K. M. (1998) A genetic algorithm for general machine scheduling problems, *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems,* Adelaide, Australia, 60 –66.

54. Lei, D. (2008) A Pareto archive particle swarm optimization for multi-objective job shop scheduling, *Computers & Industrial Engineering*, 54, 960–971.

55. Lei, D. (2009) Multi-objective production scheduling: a survey, *International Journal of Advanced Manufacturing Technology*, 43, 926-938.

56. Lei, D. (2011) Simplified multi-objective genetic algorithms for stochastic job shop scheduling, *Applied Soft Computing*, 11, 4991–4996.

57. Lei, D. and Wu, Z (2006) Crowding-measure-based multiobjective evolutionary algorithm for job shop scheduling, *International Journal of Advanced Manufacturing Technology*, 30, 112-117.

58. Lei, D. and Xiong, H (2007) An efficient evolutionary algorithm for multi-objective stochastic job shop scheduling, *Proceedings of IEEE Sixth International Conference on Machine Learning Cybernetics, Hong Kong*, 867-872.

59. Li, J., Pan Q., and Gao, K. (2011) Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems, *International Journal of Advanced Manufacturing Technology*, 55, 1159-1169.

60. Li, J., Pan, Q. and Xie, S. (2012) An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems, *Applied Mathematics and Computation*, (http://dx.doi.org/10.1016/j.amc.2012.03.018).

61. Liu, H., Abraham A., Choi, O., and Moon, S. H. (2006) Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems, *SEAL 2006, LNCS 4247,* 197–204.

62. Liu, H., Abraham, A., and Grosan, C. (2007) A novel variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems, *Proceedings of IEEE Conference,* 1-8.

63. Lobo, F. G., and Goldberg, D. E. (1997) Decision making in a hybrid genetic algorithm, *IEEE International Conference on Evolutionary Computation,* Piscataway, USA, IEEE Press, 122-125.

64. Loukil, T., Teghem, J., and Tuyttens, D. (2005) Solving multi-objective production scheduling problems using metaheuristics, *European Journal of Operational Research,* 161, 42–61.

65. Loukil, T., Teghem, J., and Fortemps, P. (2007) A multi-objective production scheduling case study solved by simulated annealing, *European Journal of Operational Research,* 179, 709–722.

66. Lu, H. L., Huang, G. Q., and Yang, H. D. (2010) Integrating order review/release and dispatching rules for assembly job shop scheduling using a simulation approach, *International Journal of Production Research,* 1-23.

67. Manikas, A. and Chang, Y. L. (2008) Multi-criteria sequence-dependent job shop scheduling using genetic algorithms, *Computers and Industrial Engineering,* 56 (1), 179-185.

68. Mckoy, D. H. C. and Egbelu, P. J. (1998) Minimizing production flow time in a process and assembly job shop, *International Journal of Production Research,* 36 (8), 2315-2332.

69. Miragliotta, G. and Perona, M. (2005) Decentralised, multi-objective driven scheduling for reentrant shops: A conceptual development and a test case, *European Journal of Operational Research,* 167, 644–662.

70. Mohanasundaram, K. M., Natarajan, K., Viswanathkumar, G., Radhakrishnan, P., and Rajendran, C. (2002) Scheduling rules for dynamic shops that manufacture multi-level jobs, *Computers and Industrial Engineering,* 44 (1), 119-131.

71. Natarajan, K., Mohanasundaram, K. M., Shoban Babu, B., Suresh, S., Raj, K. A. A. D., and Rajendran, C. (2007) Performance evaluation of priority dispatching rules in multi-level assembly job shops with jobs having weights for flowtime and tardiness, *International Journal of Advanced Manufacturing Technology,* 31, 751-761.

72. Omkumar, M., Shahabudeen, P., Gughan, S., and Azad, A. (2009) GA based static scheduling of multilevel assembly job shops, *International Journal of Operations Research,* 4 (2), 197-213.

73. Osyczka, A. and Kundu, S. (1995) A new method to solve generalized multi-criteria optimization problems using the simple genetic algorithm, *Structural Optimization,* 10 (2), 94-99.

74. Paneerselvam, R. (2007) *Design and Analysis of Algorithms,* Prentice Hall of India, New Delhi.

75. Parveen, S. and Ullah, H. (2010) Review on job-shop and flow-shop scheduling using multi criteria decision making, *Journal of Mechanical Engineering,* ME 41 (2), 130-146.

76. Pathumnakul, S. and Egbelu, P. J. (2006) An algorithm for minimizing weighted earliness penalty in assembly job shops, *International Journal of Production Economics,* 103, 230-245.

77. Pezzella, F., Morganti, G., and Ciaschetti, G. (2008) A genetic algorithm for the flexible job-shop scheduling problem, *Computers and Industrial Engineering,* 35 (10), 3202-3212.

78. Philipoom, P. R., Markland, R. E., and Fry T. D. (1989) Sequencing rules, progress milestones and product structure in a multistage job shop, *Journal of Operations Management,* 8 (3), 209-229.

79. Philipoom, P. R., Russell, R. S., and Fry, T. D. (1991) A preliminary investigation of multi attribute based sequencing rules for assembly shops, *International Journal of Production Research,* 29, 739-753.

80. Pinedo, M. L. (2005) *Planning and Scheduling in Manufacturing Systems*, Springer, New York.

81. Pongcharoen, P., Hicks, C., Braiden, P. M., and Stewardson, D. J. (2002) Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products, *International Journal of Production Economics*, 78, 311-322.

82. Pongcharoen, P., Hicks, C., and Braiden, P. M. (2004) The development of genetic algorithms for the finite capacity scheduling of complex products with multi levels of product structure, *European Journal of Operational Research*, 152 (1), 215-225.

83. Ponnambalam, S. G., Ramkumar, V., Jawahar, N. (2001) A multiobjective genetic algorithm for job shop scheduling, *Production Planning and Control,* 12 (8), 764–774.

84. Ponnambalam, S. G., Jawahar, N., and Girish, B. S. (2009) Giffler and Thompson procedure based genetic algorithms for scheduling job shops, *Comput. Intel. in Flow Shop and Job Shop Scheduling*, SCI 230, 229-259.

85. Ramesh, R. and Cary, J. M. (1989) Multicriteria jobshop scheduling, *Computers and Industrial Engineering*, 17 (1-4), 597-602.

86. Reeja, M. K. and Rajendran, C. (2000a) Dispatching rules for scheduling in assembly job shops – Part I, *International Journal of Production Research*, 38 (9), 2051-2066.

87. Reeja, M. K. and Rajendran, C. (2000b) Dispatching rules for scheduling in assembly job shops – Part II, *International Journal of Production Research*, 38 (10), 2349-2360.

88. Reeves, C. (1994) Genetic algorithms and neighbourhood search, *Evolutionary Computing, AISB Workshop, Lecture Notes in Computer Science*, T. C. Fogarty, T. C. (Ed.) Springer-Verlag, Leeds, UK, 865, 115-130.

89. Roman, D. B. and Valle, A. G. (1996) Dynamic assignation of due-dates in an assembly shop based in simulation, *International Journal of Production Research*, 34 (6), 1539-1554.

90. Sculli, D. (1980) Priority dispatching rules in job shops with assembly operations and random delays, *OMEGA The International Journal of Management Science*, 8 (2), 227-234.

91. Sculli, D. (1987) Priority dispatching rules in an assembly shop, *OMEGA The International Journal of Management Science*, 15 (1), 45-57.

92. Sha, D. Y. and Lin H. H. (2010) A multi-objective PSO for job-shop scheduling problems, *Expert Systems with Applications*, 37, 1065–1070.

93. Tavakkoli-Moghaddam, R., Azarkish, M., and Sadeghnejad-Barkousaraie, A. (2011a) Solving a multi-objective job shop scheduling problem with sequence-dependent setup times by a Pareto archive PSO combined with genetic operators and VNS, *International Journal of Advanced Manufacturing Technology*, 53, 733–750.

94. Tavakkoli-Moghaddam, R., Azarkish, M., and Sadeghnejad-Barkousaraie, A. (2011b) A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem, *Expert Systems with Applications*, 38, 10812–10821.

95. Tay, J. C. and Ho, N. B. (2008) Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems, *Computers & Industrial Engineering*, 54, 453–473.

96. Thiagarajan, S. and Rajendran, C. (2003) Scheduling in dynamic assembly job-shops with jobs having different holding and tardiness cost, *International Journal of Production Research*, 41 (18), 4453-4486.

97. Thiagarajan, S. and Rajendran, C. (2005) Scheduling in dynamic assembly job shops to minimize the sum of weighted earliness, weighted tardiness and weighted flow time of jobs, *Computers and Industrial Engineering,* 49 (4), 463-503.

98. Vilcot. G. and Billaut, J. (2008) A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem, *European Journal of Operational Research*, 190, 398–411.

99. Wang, X., Gao, L., Zhang, C. and Shao, X. (2010) A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem, *International Journal of Advanced Manufacturing Technology*, 51, 757–767.

100. Wong, T. C., Chan, F. T. S., and Chan, L. Y. (2009) A resource-constrained assembly job shop scheduling problem with lot streaming technique, *Computers & Industrial Engineering*, 57, 983–995.

101. Xia, W. and Wu, Z. (2005) An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Computers & Industrial Engineering,* 48 (2), 409–425.

102. Xing, L. N., Chen, Y. W., and Yang K. W. (2008) Multi-objective flexible job shop schedule: design and evaluation by simulation modeling, *Applied Soft Computing*, 1-15.

103. Zang, D., Shi, X., and Jin, M. (2010) Modeling and scheduling of real-life assembly job shop with timed colored petri net, *Proceedings of IEEE Conference*, 1-5.

104. Zhang, G., Shao, X., Li, P., and Gao, L. (2009) An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, *Computers & Industrial Engineering,* 56, 1309–1318.

105. Zhou, H., Feng, Y., Han, L. (2001) A hybrid heuristic genetic algorithm for job shop scheduling, *Computers & Industrial Engineering*, 40, 191-200.

106. Zitzler, E. and Thiele, L. (1998) An evolutionary algorithm for multiobjective optimization: the strength Pareto approach, *Technical Report 43*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology, Zurich, Switzerland.

107. Zitzler, E., Laumanns, M., and Thiele, L. (2001) SPEA2: improving the strength Pareto evolutionary algorithm. *Technical Report 103*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland.

108. Zitzler, E. and Thiele, L. (1999) Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*. 3, 257–271.

APPENDIX

DATA FOR THE MOAJSP INSTANCES

Table A.1 Data of problem P1

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	31	4	1	1	1	1	3	{(1, 2, 3), (1, 3, 4), (1, 4, 5)}
				2	3	1	10	5	
						2	7	4	
						3	5	10	
				3	4	1	6	8	
						2	8	2	
						3	7	5	
						4	10	1	
				4	5	1	9	11	
						2	6	2	
						3	8	4	
						4	7	3	
						5	5	8	
2	T	37	7	1	1	1	3	8	{(2, 2, 1), (2, 3, 5)}
				2	1	1	2	5	{(2, 4, 1), (2, 5, 4)}
				3	5	1	10	7	
						2	8	3	
						3	6	11	
						4	7	4	
						5	9	1	
				4	1	1	4	3	{(2, 6, 3), (2, 7, 2)}
				5	4	1	9	3	
						2	5	12	
						3	6	1	
						4	8	8	
				6	3	1	5	2	
						2	8	9	
						3	9	4	
				7	2	1	7	15	
						2	10	6	
3	C	104	12	1	1	1	3	8	{(3, 2, 2), (3, 3, 2), (3, 4, 2)}
				2	2	1	4	7	{(3, 5, 5), (3, 6, 2)}
						2	10	3	
				3	2	1	1	10	{(3, 7, 3), (3, 8, 4), (3, 9, 2)}
						2	6	2	
				4	2	1	2	6	{(3, 10, 2), (3, 11, 5), (3, 12, 2)}
						2	8	5	
				5	5	1	6	6	
						2	5	4	
						3	9	11	
						4	10	6	
						5	8	7	

Table A.1 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				6	2	1	7	10	
						2	5	3	
				7	3	1	10	8	
						2	8	3	
						3	6	12	
				8	4	1	5	10	
						2	6	1	
						3	8	13	
						4	10	2	
				9	2	1	6	5	
						2	9	1	
				10	2	1	10	11	
						2	7	5	
				11	5	1	8	15	
						2	7	3	
						3	10	8	
						4	9	2	
						5	6	4	
				12	2	1	9	8	
						2	5	14	

Table A.2 Data of problem P2

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	68	5	1	2	1	4	7	{(1, 2, 4), (1, 3, 2), (1, 4, 4), (1, 5, 3)}
						2	5	3	
				2	4	1	9	14	
						2	7	7	
						3	8	6	
						4	5	8	
				3	2	1	6	15	
						2	7	4	
				4	4	1	5	2	
						2	6	5	
						3	10	13	
						4	8	4	
				5	3	1	8	9	
						2	7	8	
						3	9	2	
2	T	56	7	1	2	1	2	4	{(2, 2, 1), (2, 3, 4)}
						2	6	3	
				2	1	1	1	9	{(2, 4, 2), (2, 5, 5)}
				3	4	1	7	4	
						2	9	3	
						3	10	12	
						4	5	8	
				4	2	1	3	11	{(2, 6, 3), (2, 7, 3)}
						2	9	7	

Table A.2 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				5	5	1	6	5	
						2	10	3	
						3	8	8	
						4	7	10	
						5	9	6	
				6	3	1	10	8	
						2	5	7	
						3	7	4	
				7	3	1	8	14	
						2	9	6	
						3	5	2	
3	C	65	16	1	1	1	3	10	{(3, 2, 1), (3, 3, 1), (3, 4, 1), (3,5,1)}
				2	1	1	4	5	{(3, 6, 2), (3, 7, 3)}
				3	1	1	1	9	{(3, 8, 3), (3, 9, 2), (3, 10, 4)}
				4	1	1	2	7	{(3, 11, 2), (3, 12, 3), (3, 13, 3), (3, 14, 4)}
				5	1	1	3	4	{(3, 15, 5), (3, 16, 2)}
				6	2	1	9	14	
						2	5	3	
				7	3	1	10	3	
						2	8	1	
						3	7	10	
				8	3	1	7	2	
						2	9	5	
						3	6	10	
				9	2	1	8	15	
						2	5	3	
				10	4	1	9	2	
						2	5	11	
						3	8	7	
						4	10	4	
				11	2	1	10	6	
							6	12	
				12	3	1	7	13	
							9	1	
							5	4	
				13	3	1	10	5	
						2	6	13	
						3	9	3	
				14	4	1	8	1	
						2	5	13	
						3	10	4	
						4	7	2	
				15	5	1	5	3	
						2	9	7	
						3	10	2	
						4	8	9	
						5	6	8	
				16	2	1	6	8	
						2	7	1	

Table A.3 Data of problem P3

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	48	7	1	1	1	1	5	{(1, 2, 3), (1, 3, 4), (1, 4, 3), (1, 5, 5), (1, 6, 4), (1, 7, 2)}
				2	3	1	8	5	
						2	5	8	
						3	6	3	
				3	4	1	6	8	
						2	10	7	
						3	9	6	
						4	5	2	
				4	3	1	9	5	
						2	7	2	
						3	8	9	
				5	5	1	6	2	
						2	7	13	
						3	8	1	
						4	5	4	
						5	10	7	
				6	4	1	5	3	
						2	6	7	
						3	10	6	
						4	9	2	
				7	2	1	7	11	
						2	5	5	
2	T	61	11	1	1	1	1	10	{(2, 2, 1), (2, 3, 3)}
				2	1	1	2	5	{(2, 4, 1), (2, 5, 3)}
				3	3	1	9	1	
						2	10	2	
						3	5	13	
				4	1	1	3	8	{(2, 6, 1), (2, 7, 5)}
				5	3	1	6	5	
						2	9	14	
						3	7	3	
				6	1	1	4	7	{(2, 8, 1), (2, 9, 2)}
				7	5	1	10	11	
						2	5	8	
						3	7	9	
						4	9	4	
						5	8	6	
				8	1	1	1	6	{(2, 10, 3), (2, 11, 4)}
				9	2	1	6	15	
						2	5	9	
				10	3	1	7	5	
						2	8	11	
						3	10	1	
				11	4	1	5	3	
						2	7	5	
						3	6	4	
						4	9	7	
3	C	72	14	1	2	1	1	9	{(3, 2, 1), (3, 3, 1), (3, 4, 1)}
						2	8	4	

Table A.3 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				2	1	1	2	8	{(3, 5, 2), (3, 6, 5), (3, 7, 3)}
				3	1	1	3	10	{(3, 8, 3), (3, 9, 4), (3, 10, 2)}
				4	1	1	4	6	{(3, 11, 4), (3, 12, 3), (3, 13, 2), (3, 14, 2)}
				5	2	1	9	10	
						2	5	4	
				6	5	1	5	7	
						2	10	3	
						3	7	1	
						4	8	5	
						5	9	2	
				7	3	1	10	6	
						2	7	12	
						3	6	3	
				8	3	1	6	15	
						2	9	4	
						3	8	1	
				9	4	1	5	10	
						2	7	3	
						3	10	8	
						4	6	2	
				10	2	1	8	4	
						2	5	10	
				11	4	1	7	9	
						2	10	2	
						3	8	14	
						4	7	4	
				12	3	1	5	9	
						2	9	3	
						3	10	1	
				13	2	1	8	7	
						2	9	11	
				14	2	1	10	13	
							7	8	

Table A.4 Data of problem P4

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	88	8	1	2	1	1	7	{(1, 2, 2), (1, 3, 4), (1, 4, 3), (1, 5, 5), (1, 6, 5), (1, 7, 3), (1, 8, 3)}
						2	5	5	
				2	2	1	5	2	
						2	8	6	
				3	4	1	9	1	
						2	10	7	
						3	6	9	
						4	7	3	
				4	3	1	6	2	
						2	5	3	

Table A.4 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	7	12	
				5	5	1	10	2	
						2	9	8	
						3	8	6	
						4	7	7	
						5	6	9	
				6	5	1	6	3	
						2	5	12	
						3	7	6	
						4	8	3	
						5	9	1	
				7	3	1	10	15	
						2	6	2	
						3	9	1	
				8	3	1	5	4	
						2	9	7	
						3	7	8	
2	T	51	7	1	1	1	1	10	{(2, 2, 2), (2, 3, 3)}
				2	2	1	3	7	{(2, 4, 2), (2, 5, 2)}
						2	8	4	
				3	3	1	5	8	
						2	6	9	
						3	10	2	
				4	2	1	4	6	{(2, 6, 4), (2, 7, 3)}
						2	7	5	
				5	2	1	10	11	
						2	7	3	
				6	4	1	5	2	
						2	7	12	
						3	9	1	
						4	6	4	
				7	3	1	9	7	
						2	10	8	
						3	6	3	
3	C	48	16	1	1	1	2	13	{(3, 2, 2), (3, 3, 2), (3, 4, 2), (3, 5, 2)}
				2	2	1	3	10	{(3, 6, 2), (3, 7, 3), (3, 8, 2)}
						2	9	4	
				3	2	1	4	5	{(3, 9, 3), (3, 10, 5)}
						2	10	5	
				4	2	1	2	4	{(3, 11, 4), (3, 12, 3)}
						2	5	6	
				5	2	1	1	11	{(3, 13, 2), (3, 14, 3), (3, 15, 2), (3, 16, 3)}
						2	7	4	
				6	2	1	10	9	
						2	6	10	
				7	3	1	5	11	
						2	8	9	
						3	9	1	

Table A.4 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				8	2	1	7	2	
						2	10	8	
				9	3	1	7	14	
						2	6	3	
						3	9	6	
				10	5	1	8	5	
						2	6	7	
						3	9	4	
						4	5	8	
						5	10	1	
				11	4	1	5	1	
						2	9	3	
						3	10	15	
						4	7	4	
				12	3	1	9	7	
						2	6	8	
						3	5	4	
				13	2	1	10	5	
						2	8	7	
				14	3	1	10	5	
						2	9	13	
						3	5	2	
				15	2	1	8	1	
						2	7	12	
				16	3	1	6	10	
						2	10	4	
						3	5	6	

Table A.5 Data of problem P5

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	48	11	1	1	1	3	12	{(1, 2, 3), (1, 3, 4), (1, 4, 5), (1, 5, 5), (1, 6, 3), (1, 7, 2), (1, 8, 2), (1, 9, 4), (1, 10, 3), (1, 11, 2)}
				2	3	1	5	15	
						2	7	4	
						3	9	1	
				3	4	1	5	5	
						2	10	3	
						3	6	14	
						4	8	2	
				4	5	1	8	6	
						2	6	3	
						3	5	9	
						4	7	11	
						5	10	1	
				5	5	1	7	4	
						2	5	9	
						3	9	7	

Table A.5 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	6	13	
						5	8	3	
				6	3	1	7	7	
						2	9	2	
						3	5	9	
				7	2	1	9	12	
						2	10	6	
				8	2	1	10	2	
						2	7	10	
				9	4	1	8	7	
						2	7	13	
						3	6	8	
						4	9	3	
				10	3	1	6	2	
						2	5	10	
						3	7	9	
				11	2	1	5	9	
						2	6	8	
2	T	41	5	1	2	1	2	9	{(2, 2, 1), (2, 3, 4)}
						2	5	4	
				2	1	1	3	6	{(2, 4, 3), (2, 5, 2)}
				3	4	1	9	8	
						2	6	4	
						3	10	13	
						4	7	3	
				4	3	1	9	5	
						2	7	14	
						3	6	1	
				5	2	1	8	9	
						2	5	3	
3	C	89	22	1	1	1	2	9	{(3, 2, 1), (3, 3, 1)}
				2	1	1	3	5	{(3, 4, 1), (3, 5, 1), (3, 6, 1)}
				3	1	1	4	8	{(3, 7, 1), (3, 8, 1), (3, 9, 1)}
				4	1	1	1	6	{(3, 10, 3), (3, 11, 3)}
				5	1	1	3	10	{(3, 12, 4), (3, 13, 2)}
				6	1	1	2	3	{(3, 14, 3), (3, 15, 2)}
				7	1	1	4	6	{(3, 16, 3), (3, 17, 2), (3, 18, 3)}
				8	1	1	1	5	{(3, 19, 5), (3, 20, 2)}
				9	1	1	2	7	{(3, 21, 2), (3, 22, 2)}
				10	3	1	8	12	
						2	6	1	
						3	9	7	
				11	3	1	7	11	
						2	10	1	
						3	5	3	
				12	4	1	5	13	
						2	9	5	
						3	8	2	
						4	6	3	
				13	2	1	10	12	

Table A.5 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	7	4	
				14	3	1	6	8	
						2	5	14	
						3	8	5	
				15	2	1	9	10	
						2	10	3	
				16	3	1	10	8	
						2	5	4	
						3	8	13	
				17	2	1	5	11	
						2	6	5	
				18	3	1	10	6	
						2	9	9	
						3	7	8	
				19	5	1	7	4	
						2	6	15	
						3	9	3	
						4	10	8	
						5	5	7	
				20	2	1	10	5	
						2	7	10	
				21	2	1	8	9	
						2	6	3	
				22	2	1	9	13	
						2	5	2	

Table A.6 Data of problem P6

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	74	11	1	2	1	4	14	{(1, 2, 2), (1, 3, 4), (1, 4, 3), (1, 5, 3), (1, 6, 4), (1, 7, 2), (1, 8, 5), (1, 9, 4), (1, 10, 2), (1, 11, 2)}
						2	7	2	
				2	2	1	5	11	
						2	10	9	
				3	4	1	9	6	
						2	8	5	
						3	6	10	
						4	7	2	
				4	3	1	7	8	
						2	10	4	
						3	6	12	
				5	3	1	5	3	
						2	8	15	
						3	9	9	
				6	4	1	10	4	
						2	9	15	
						3	7	2	
						4	8	1	

Table A.6 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				7	2	1	10	14	
						2	6	1	
				8	5	1	5	6	
						2	6	9	
						3	9	1	
						4	7	14	
						5	10	3	
				9	4	1	9	8	
						2	8	10	
						3	7	1	
						4	5	13	
				10	2	1	8	3	
						2	5	8	
				11	2	1	6	9	
						2	7	2	
2	T	84	7	1	2	1	2	4	{(2, 2, 1), (2, 3, 4)}
						2	6	3	
				2	1	1	1	9	{(2, 4, 2), (2, 5, 5)}
				3	4	1	7	4	
						2	9	3	
						3	10	12	
						4	5	8	
				4	2	1	3	11	{(2, 6, 3), (2, 7, 3)}
						2	9	7	
				5	5	1	6	5	
						2	10	3	
						3	8	8	
						4	7	10	
						5	9	6	
				6	3	1	10	8	
						2	5	7	
						3	7	4	
				7	3	1	8	14	
						2	9	6	
						3	5	2	
3	C	59	20	1	2	1	2	11	{(3, 2, 1), (3, 3, 2)}
						2	5	4	
				2	1	1	3	10	{(3, 4, 1), (3, 5, 1), (3, 6, 1)}
				3	2	1	4	8	{(3, 7, 1), (3, 8, 1)}
						2	8	5	
				4	1	1	1	3	{(3, 9, 3), (3, 10, 4)}
				5	1	1	4	8	{(3, 11, 2), (3, 12, 5)}
				6	1	1	3	5	{(3, 13, 3), (3, 14, 2)}
				7	1	1	2	9	{(3, 15, 2), (3,16, 2), (3, 17, 3)}
				8	1	1	3	4	{(3, 18, 2), (3, 19, 4), (3, 20, 3)}
				9	3	1	5	3	
						2	7	6	
						3	8	4	
				10	4	1	6	8	
						2	10	2	

Table A.6 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	5	2	
						4	9	10	
				11	2	1	10	2	
						2	6	12	
				12	5	1	9	1	
						2	5	10	
						3	6	3	
						4	7	8	
						5	8	4	
				13	3	1	8	15	
						2	10	2	
						3	5	1	
				14	2	1	6	9	
						2	9	8	
				15	2	1	7	15	
						2	8	3	
				16	2	1	5	2	
						2	9	14	
				17	3	1	10	3	
						2	6	13	
						3	8	5	
				18	2	1	5	4	
						2	9	10	
				19	4	1	6	7	
						2	8	2	
						3	5	11	
						4	10	6	
				20	3	1	9	5	
						2	10	14	
						3	6	1	

Table A.7 Data of problem P7

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	48	7	1	1	1	1	5	{(1, 2, 3), (1, 3, 4), (1, 4, 3), (1, 5, 5), (1, 6, 4), (1, 7, 2)}
				2	3	1	8	5	
						2	5	8	
						3	6	3	
				3	4	1	6	8	
						2	10	7	
						3	9	6	
						4	5	2	
				4	3	1	9	5	
						2	7	2	
						3	8	9	
				5	5	1	6	2	
						2	7	13	
						3	8	1	

Table A.7 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	5	4	
						5	10	7	
				6	4	1	5	3	
						2	6	7	
						3	10	6	
						4	9	2	
				7	2	1	7	11	
						2	5	5	
2	T	40	5	1	1	1	1	5	{(2, 2, 1), (2, 3, 4)}
				2	1	1	2	7	{(2, 4, 5), (2, 5, 3)}
				3	4	1	10	11	
						2	6	3	
						3	9	5	
						4	7	2	
				4	5	1	6	9	
						2	5	4	
						3	9	6	
						4	8	7	
						5	10	2	
				5	3	1	7	12	
						2	8	3	
						3	5	1	
3	T	62	5	1	2	1	2	9	{(3, 2, 1), (3, 3, 4)}
						2	5	4	
				2	1	1	3	6	{(3, 4, 3), (3, 5, 2)}
				3	4	1	9	8	
						2	6	4	
						3	10	13	
						4	7	3	
				4	3	1	9	5	
						2	7	14	
						3	6	1	
				5	2	1	8	9	
						2	5	3	
4	T	150	9	1	2	1	3	12	{(4, 2, 1), (4, 3, 2)}
						2	9	4	
				2	1	1	1	9	{(4, 4, 2), (4, 5, 3)}
				3	2	1	8	7	
						2	5	9	
				4	2	1	3	13	{(4, 6, 1), (4, 7, 5)}
						2	5	5	
				5	3	1	5	4	
						2	10	9	
						3	6	6	
				6	1	1	2	8	{(4, 8, 3), (4, 9, 3)}
				7	5	1	10	9	
						2	6	5	
						3	8	8	
						4	5	6	
						5	9	4	

Table A.7 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				8	3	1	10	2	
						2	7	11	
						3	5	7	
				9	3	1	7	3	
						2	8	7	
						3	6	10	
5	C	48	14	1	2	1	1	9	{(5, 2, 1), (5, 3, 1), (5, 4, 1)}
						2	8	4	
				2	1	1	2	8	{(5, 5, 2), (5, 6, 5), (5, 7, 3)}
				3	1	1	3	10	{(5, 8, 3), (5, 9, 4), (5, 10, 2)}
				4	1	1	4	6	{(5, 11, 4), (5, 12, 3), (5, 13, 2), (5, 14, 2)}
				5	2	1	9	10	
						2	5	4	
				6	5	1	5	7	
						2	10	3	
						3	7	1	
						4	8	5	
						5	9	2	
				7	3	1	10	6	
						2	7	12	
						3	6	3	
				8	3	1	6	15	
						2	9	4	
						3	8	1	
				9	4	1	5	10	
						2	7	3	
						3	10	8	
						4	6	2	
				10	2	1	8	4	
						2	5	10	
				11	4	1	7	9	
						2	10	2	
						3	8	14	
						4	7	4	
				12	3	1	5	9	
						2	9	3	
						3	10	1	
				13	2	1	8	7	
						2	9	11	
				14	2	1	10	13	
							7	8	

Table A.8 Data of problem P8

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	88	7	1	2	1	4	9	{(1, 2, 4), (1, 3, 2), (1, 4, 4), (1, 5, 3), (1, 6, 2), (1, 7, 5)}
						2	5	11	

Table A.8 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				2	4	1	6	1	
						2	5	12	
						3	7	2	
						4	9	3	
				3	2	1	8	13	
						2	6	1	
				4	4	1	5	9	
						2	9	3	
						3	10	8	
						4	6	4	
				5	3	1	5	4	
						2	9	3	
						3	10	6	
				6	2	1	7	14	
						2	8	1	
				7	5	1	10	1	
						2	5	3	
						3	6	7	
						4	7	8	
						5	8	3	
2	T	45	5	1	2	1	4	10	{(2, 2, 2), (2, 3, 4)}
						2	9	5	
				2	2	1	1	4	{(2, 4, 4), (2, 5, 3)}
						2	8	3	
				3	4	1	5	2	
						2	7	15	
						3	9	2	
						4	10	1	
				4	4	1	9	9	
						2	6	2	
						3	8	4	
						4	5	8	
				5	3	1	10	7	
						2	7	8	
						3	9	3	
3	T	77	7	1	1	1	1	10	{(3, 2, 2), (3, 3, 3)}
				2	2	1	3	7	{(3, 4, 2), (3, 5, 2)}
						2	8	4	
				3	3	1	5	8	
						2	6	9	
						3	10	2	
				4	2	1	4	6	{(3, 6, 4), (3, 7, 3)}
						2	7	5	
				5	2	1	10	11	
						2	7	3	
				6	4	1	5	2	
						2	7	12	
						3	9	1	
						4	6	4	
				7	3	1	9	7	

Table A.8 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	10	8	
						3	6	3	
4	T	101	9	1	1	1	4	12	{(4, 2, 2), (4, 3, 3)}
				2	2	1	2	5	{(4, 4, 2), (4, 5, 2)}
						2	7	7	
				3	3	1	5	4	
						2	6	9	
						3	8	1	
				4	2	1	4	9	{(4, 6, 2), (4, 7, 3)}
						2	6	2	
				5	2	1	10	10	
						2	9	5	
				6	2	1	1	4	{(4, 8, 3), (4, 9, 4)}
						2	9	3	
				7	3	1	6	15	
						2	9	6	
						3	5	1	
				8	3	1	5	7	
						2	9	2	
						3	8	13	
				9	4	1	9	2	
						2	10	12	
						3	6	6	
						4	7	5	
5	C	48	13	1	2	1	4	12	{(5, 2, 1), (5, 3, 2), (5, 4, 1)}
						2	10	3	
				2	1	1	1	9	{(5, 5, 3), (5, 6, 4), (5, 7, 2), (5, 8, 3)}
				3	2	1	2	3	{(5, 9, 2), (5, 10, 5)}
						2	6	4	
				4	1	1	3	7	{(5, 11, 3), (5, 12, 2), (5, 13, 4)}
				5	3	1	9	13	
						2	6	4	
						3	10	1	
				6	4	1	5	4	
						2	6	3	
						3	8	2	
						4	7	12	
				7	2	1	8	9	
						2	10	4	
				8	3	1	7	11	
						2	6	2	
						3	5	6	
				9	2	1	6	15	
						2	9	5	
				10	5	1	10	1	
						2	9	7	
						3	8	3	
						4	5	10	
						5	7	5	

Table A.8 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				11	3	1	9	3	
						2	8	14	
						3	5	4	
				12	2	1	10	10	
						2	7	8	
				13	4	1	8	5	
						2	6	2	
						3	9	11	
						4	5	6	

Table A.9 Data of problem P9

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	32	5	1	1	1	3	8	{(1, 2, 3), (1, 3, 4), (1, 4, 3), (1, 5, 5)}
				2	3	1	5	2	
						2	6	13	
						3	9	6	
				3	4	1	9	4	
						2	7	1	
						3	8	3	
						4	6	9	
				4	3	1	8	5	
						2	10	3	
						3	5	11	
				5	5	1	10	2	
						2	7	8	
						3	8	10	
						4	5	1	
						5	6	3	
2	F	53	9	1	1	1	2	9	{(2, 2, 5), (2, 3, 4), (2, 4, 2), (2, 5, 3), (2, 6, 4), (2, 7, 5), (2, 8, 2), (2, 9, 3)}
				2	5	1	8	3	
						2	9	2	
						3	10	6	
						4	5	8	
						5	7	3	
				3	4	1	6	1	
						2	7	4	
						3	5	2	
						4	9	10	
				4	2	1	6	11	
						2	5	9	
				5	3	1	5	2	
						2	7	7	
						3	9	5	
				6	4	1	6	2	
						2	5	4	

Table A.9 (Continued)

p	*PT*	D_p	n_p	*i*	J_{pi}	*j*	*m*	t_{pijm}	E_{pij}
						3	10	13	
						4	7	1	
				7	5	1	8	2	
						2	9	7	
						3	10	3	
						4	5	6	
						5	7	8	
				8	2	1	10	1	
						2	8	15	
				9	3	1	7	9	
						2	5	7	
						3	9	5	
3	T	40	5	1	1	1	1	5	{(3, 2, 1), (3, 3, 4)}
				2	1	1	2	7	{(3, 4, 5), (3, 5, 3)}
				3	4	1	10	11	
						2	6	3	
						3	9	5	
						4	7	2	
				4	5	1	6	9	
						2	5	4	
						3	9	6	
						4	8	7	
						5	10	2	
				5	3	1	7	12	
						2	8	3	
						3	5	1	
4	T	78	7	1	2	1	2	8	{(4, 2, 2), (4, 3, 3)}
						2	8	2	
				2	2	1	4	7	{(4, 4, 1), (4, 5, 3)}
						2	9	3	
				3	3	1	10	3	
						2	5	5	
						3	6	7	
				4	1	1	1	5	{(4, 6, 2), (4, 7, 4)}
				5	3	1	7	4	
						2	8	13	
						3	9	1	
				6	2	1	5	9	
						2	10	5	
				7	4	1	9	1	
						2	5	7	
						3	7	15	
						4	8	4	
5	C	106	17	1	2	1	3	15	{(5, 2, 2), (5, 3, 1), (5, 4, 2), (5, 5,1)}
						2	10	2	
				2	2	1	2	4	{(5, 6, 5), (5, 7, 3)}
						2	7	3	
				3	1	1	1	7	{(5, 8, 3), (5, 9, 2), (5, 10, 2)}

Table A.9 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				4	2	1	4	9	{(5, 11, 3), (5, 12, 2), (5, 13, 3), (5, 14, 2)}
						2	8	5	
				5	1	1	3	6	{(5, 15, 3), (5, 16, 4), (5, 17, 2)}
				6	5	1	9	8	
						2	8	4	
						3	5	10	
						4	10	1	
						5	6	3	
				7	3	1	6	7	
						2	7	10	
						3	9	1	
				8	3	1	5	9	
						2	10	2	
						3	8	4	
				9	2	1	7	8	
						2	6	11	
				10	2	1	8	12	
						2	9	7	
				11	3	1	10	13	
						2	6	4	
						3	5	5	
				12	2	1	5	4	
						2	10	9	
				13	3	1	9	13	
						2	8	3	
						3	7	6	
				14	2	1	6	8	
						2	5	6	
				15	3	1	5	14	
						2	9	5	
						3	8	3	
				16	4	1	7	2	
						2	10	9	
						3	6	5	
						4	9	1	
				17	2	1	10	6	
						2	7	15	

Table A.10 Data of problem P10

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	45	5	1	2	1	4	7	{(1, 2, 4), (1, 3, 2), (1, 4, 4), (1, 5, 3)}
						2	5	3	
				2	4	1	9	14	
						2	7	7	
						3	8	6	

Table A.10 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	5	8	
				3	2	1	6	15	
						2	7	4	
				4	4	1	5	2	
						2	6	5	
						3	10	13	
						4	8	4	
				5	3	1	8	9	
						2	7	8	
						3	9	2	
2	F	66	9	1	2	1	4	8	{(2, 2, 2), (2, 3, 5), (2, 4, 3), (2, 5, 3), (2, 6, 4), (2, 7, 2), (2, 8, 4), (2, 9, 5)}
						2	5	7	
				2	2	1	7	13	
						2	6	6	
				3	5	1	10	6	
						2	8	3	
						3	5	10	
						4	7	5	
						5	9	3	
				4	3	1	7	13	
						2	6	8	
						3	8	6	
				5	3	1	9	11	
						2	7	7	
						3	6	4	
				6	4	1	9	2	
						2	8	13	
						3	10	1	
						4	5	5	
				7	2	1	10	15	
						2	7	3	
				8	4	1	10	7	
						2	6	14	
						3	9	5	
						4	8	1	
				9	5	1	8	3	
						2	9	9	
						3	5	4	
						4	7	8	
						5	10	5	
3	T	56	5	1	1	1	3	8	{(3, 2, 2), (3, 3, 5)}
				2	2	1	4	8	{(3, 4, 2), (3, 5, 3)}
						2	6	3	
				3	5	1	9	2	
						2	5	9	
						3	10	4	
						4	7	6	
						5	8	1	

Table A.10 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				4	2	1	8	12	
						2	9	5	
				5	3	1	6	8	
						2	5	7	
						3	10	3	
4	T	112	7	1	2	1	2	4	{(4, 2, 1), (4, 3, 4)}
						2	6	3	
				2	1	1	1	9	{(4, 4, 2), (4, 5, 5)}
				3	4	1	7	4	
						2	9	3	
						3	10	12	
						4	5	8	
				4	2	1	3	11	{(4, 6, 3), (4, 7, 3)}
						2	9	7	
				5	5	1	6	5	
						2	10	3	
						3	8	8	
						4	7	10	
						5	9	6	
				6	3	1	10	8	
						2	5	7	
						3	7	4	
				7	3	1	8	14	
						2	9	6	
						3	5	2	
5	C	53	17	1	2	1	4	10	{(5, 2, 1), (5, 3, 2), (5, 4, 2), (5, 5, 2)}
						2	9	5	
				2	1	1	1	8	{(5, 6, 2), (5, 7, 3), (5, 8, 3)}
				3	2	1	2	6	{(5, 9, 3), (5, 10, 4), (5, 11, 2)}
						2	7	3	
				4	2	1	4	11	{(5, 12, 2), (5, 13, 2), (5, 14, 3), (5, 15, 4)}
						2	5	2	
				5	2	1	3	3	{(5, 16, 5), (5, 17, 2)}
						2	6	4	
				6	2	1	5	10	
						2	10	2	
				7	3	1	6	3	
						2	9	15	
						3	8	1	
				8	3	1	10	8	
						2	5	2	
						3	7	5	
				9	3	1	8	1	
						2	9	3	
						3	6	12	
				10	4	1	7	12	
						2	10	4	
						3	8	5	

Table A.10 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	5	1	
				11	2	1	10	14	
						2	7	6	
				12	2	1	9	14	
						2	5	4	
				13	2	1	6	4	
						2	7	13	
				14	3	1	5	10	
						2	9	8	
						3	7	3	
				15	4	1	10	6	
						2	7	4	
						3	8	11	
						4	6	1	
				16	5	1	8	2	
						2	10	7	
						3	9	3	
						4	5	10	
						5	6	9	
				17	2	1	9	6	
						2	7	8	

Table A.11 Data of problem P11

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	70	6	1	1	1	2	12	{(1, 2, 5), (1, 3, 3), (1, 4, 2), (1, 5, 3), (1, 6, 4)}
				2	5	1	8	4	
						2	9	2	
						3	5	8	
						4	6	3	
						5	10	3	
				3	3	1	9	5	
						2	8	13	
						3	5	1	
				4	2	1	8	15	
						2	10	2	
				5	3	1	7	8	
						2	5	11	
						3	6	2	
				6	4	1	5	6	
						2	10	9	
						3	8	5	
						4	7	3	
2	T	60	9	1	1	1	1	9	{(2, 2, 1), (2, 3, 4)}
				2	1	1	3	4	{(2, 4, 1), (2, 5, 2)}
				3	4	1	9	15	
						2	8	2	
						3	5	3	

Table A.11 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	6	7	
				4	1	1	4	5	{(2, 6, 1), (2, 7, 3)}
				5	2	1	7	9	
						2	10	6	
				6	1	1	1	7	{(2, 8, 2), (2, 9, 3)}
				7	3	1	5	4	
						2	8	14	
						3	9	1	
				8	2	1	10	13	
						2	5	2	
				9	3	1	7	3	
						2	8	1	
						3	6	11	
3	C	44	12	1	1	1	1	9	{(3, 2, 1), (3, 3, 1), (3, 4, 1)}
				2	1	1	2	7	{(3, 5, 3), (3, 6, 2), (3, 7, 4)}
				3	1	1	3	5	{(3, 8, 3), (3, 9, 5)}
				4	1	1	4	10	{(3, 10, 2), (3, 11, 3), (3, 12, 2)}
				5	3	1	5	11	
						2	7	2	
						3	8	4	
				6	2	1	6	5	
						2	10	8	
				7	4	1	10	3	
						2	7	13	
						3	9	1	
						4	5	5	
				8	3	1	8	6	
						2	9	7	
						3	6	12	
				9	5	1	6	8	
						2	8	7	
						3	7	2	
						4	10	3	
						5	9	10	
				10	2	1	8	15	
						2	9	3	
				11	3	1	9	4	
						2	10	14	
						3	7	3	
				12	2	1	7	6	
						2	5	9	
4	C	75	15	1	1	1	4	13	{(4, 2, 1), (4, 3, 1), (4, 4, 1), (4, 5, 1)}
				2	1	1	1	6	{(4, 6, 5), (4, 7, 3)}
				3	1	1	2	4	{(4, 8, 2), (4, 9, 3)}
				4	1	1	3	8	{(4, 10, 5), (4, 11, 2), (4, 12, 3)}
				5	1	1	4	9	{(4, 13, 4), (4, 14, 2), (4, 15, 2)}
				6	5	1	10	1	
						2	9	3	
						3	7	8	

Table A.11 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	5	15	
						5	8	2	
				7	3	1	7	4	
						2	10	12	
						3	6	5	
				8	2	1	8	4	
						2	9	13	
				9	3	1	10	7	
						2	5	5	
						3	7	11	
				10	5	1	5	11	
						2	6	4	
						3	8	5	
						4	9	8	
						5	7	1	
				11	2	1	6	2	
						2	8	10	
				12	3	1	7	4	
						2	10	5	
						3	8	4	
				13	4	1	5	4	
						2	10	6	
						3	8	4	
						4	6	3	
				14	2	1	9	10	
						2	5	9	
				15	2	1	6	8	
						2	7	3	
5	C	118	22	1	1	1	2	9	{(5, 2, 1), (5, 3, 1)}
				2	1	1	3	5	{(5, 4, 1), (5, 5, 1), (5, 6, 1)}
				3	1	1	4	8	{(5, 7, 1), (5, 8, 1), (5, 9, 1)}
				4	1	1	1	6	{(5, 10, 3), (5, 11, 3)}
				5	1	1	3	10	{(5, 12, 4), (5, 13, 2)}
				6	1	1	2	3	{(5, 14, 3), (5, 15, 2)}
				7	1	1	4	6	{(5, 16, 3), (5, 17, 2), (5, 18, 3)}
				8	1	1	1	5	{(5, 19, 5), (5, 20, 2)}
				9	1	1	2	7	{(5, 21, 2), (5, 22, 2)}
				10	3	1	8	12	
						2	6	1	
						3	9	7	
				11	3	1	7	11	
						2	10	1	
						3	5	3	
				12	4	1	5	13	
						2	9	5	
						3	8	2	
						4	6	3	
				13	2	1	10	12	
						2	7	4	
				14	3	1	6	8	

Table A.11 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	5	14	
						3	8	5	
				15	2	1	9	10	
						2	10	3	
				16	3	1	10	8	
						2	5	4	
						3	8	13	
				17	2	1	5	11	
						2	6	5	
				18	3	1	10	6	
						2	9	9	
						3	7	8	
				19	5	1	7	4	
						2	6	15	
						3	9	3	
						4	10	8	
						5	5	7	
				20	2	1	10	5	
						2	7	10	
				21	2	1	8	9	
						2	6	3	
				22	2	1	9	13	
						2	5	2	

Table A.12 Data of problem P12

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	68	6	1	2	1	3	7	{(1, 2, 2), (1, 3, 4), (1, 4, 3), (1, 5, 5), (1, 6, 2)}
						2	10	4	
				2	2	1	8	15	
						2	7	2	
				3	4	1	9	2	
						2	8	9	
						3	5	6	
						4	6	3	
				4	3	1	10	1	
						2	7	10	
						3	8	5	
				5	5	1	5	4	
						2	6	3	
						3	8	7	
						4	9	8	
						5	10	1	
				6	2	1	6	11	
						2	7	3	
2	T	64	9	1	1	1	2	11	{(2, 2, 2), (2, 3, 5)}
				2	2	1	4	8	{(2, 4, 1), (2, 5, 3)}
						2	8	3	

Table A.12 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				3	5	1	5	1	
						2	9	9	
						3	6	5	
						4	7	4	
						5	10	8	
				4	1	1	3	6	{(2, 6, 2), (2, 7, 2)}
				5	3	1	6	7	
						2	5	11	
						3	9	2	
				6	2	1	2	7	{(2, 8, 4), (2, 9, 2)}
						2	7	6	
				7	2	1	7	10	
						2	8	5	
				8	4	1	8	5	
						2	10	13	
						3	7	2	
						4	5	3	
				9	2	1	6	8	
						2	8	5	
3	C	72	12	1	1	1	2	12	{(3, 2, 1), (3, 3, 1), (3, 4, 1)}
				2	1	1	3	8	{(3, 5, 3), (3, 6, 4)}
				3	1	1	4	6	{(3, 7, 3), (3, 8, 3)}
				4	1	1	1	7	{(3, 9, 2), (3, 10, 2), (3, 11, 4), (3, 12, 5)}
				5	3	1	9	3	
						2	6	8	
						3	8	7	
				6	4	1	7	1	
						2	10	6	
						3	9	10	
						4	5	9	
				7	3	1	5	2	
						2	7	13	
						3	8	4	
				8	3	1	8	3	
						2	6	11	
						3	10	5	
				9	2	1	7	10	
						2	5	1	
				10	2	1	9	4	
						2	5	12	
				11	4	1	6	3	
						2	8	14	
						3	9	1	
						4	7	4	
				12	5	1	5	5	
						2	7	4	
						3	6	2	
						4	10	15	
						5	9	3	

Table A.12 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
4	C	96	16	1	1	1	2	13	{(4, 2, 2), (4, 3, 2), (4, 4, 2), (4, 5, 2)}
				2	2	1	3	10	{(4, 6, 2), (4, 7, 3), (4, 8, 2)}
						2	9	4	
				3	2	1	4	5	{(4, 9, 3), (4, 10, 5)}
						2	10	5	
				4	2	1	2	4	{(4, 11, 4), (4, 12, 3)}
						2	5	6	
				5	2	1	1	11	{(4, 13, 2), (4, 14, 3), (4, 15, 2), (4, 16, 3)}
						2	7	4	
				6	2	1	10	9	
						2	6	10	
				7	3	1	5	11	
						2	8	9	
						3	9	1	
				8	2	1	7	2	
						2	10	8	
				9	3	1	7	14	
						2	6	3	
						3	9	6	
				10	5	1	8	5	
						2	6	7	
						3	9	4	
						4	5	8	
						5	10	1	
				11	4	1	5	1	
						2	9	3	
						3	10	15	
						4	7	4	
				12	3	1	9	7	
						2	6	8	
						3	5	4	
				13	2	1	10	5	
						2	8	7	
				14	3	1	10	5	
						2	9	13	
						3	5	2	
				15	2	1	8	1	
						2	7	12	
				16	3	1	6	10	
						2	10	4	
						3	5	6	
5	C	118	20	1	2	1	2	11	{(5, 2, 1), (5, 3, 2)}
						2	5	4	
				2	1	1	3	10	{(5, 4, 1), (5, 5, 1), (5, 6, 1)}
				3	2	1	4	8	{(5, 7, 1), (5, 8, 1)}
						2	8	5	
				4	1	1	1	3	{(5, 9, 3), (5, 10, 4)}
				5	1	1	4	8	{(5, 11, 2), (5, 12, 5)}

Table A.12 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				6	1	1	3	5	{(5, 13, 3), (5, 14, 2)}
				7	1	1	2	9	{(5, 15, 2), (5,16, 2), (5, 17, 3)}
				8	1	1	3	4	{(5, 18, 2), (5, 19, 4), (5, 20, 3)}
				9	3	1	5	3	
						2	7	6	
						3	8	4	
				10	4	1	6	8	
						2	10	2	
						3	5	2	
						4	9	10	
				11	2	1	10	2	
						2	6	12	
				12	5	1	9	1	
						2	5	10	
						3	6	3	
						4	7	8	
						5	8	4	
				13	3	1	8	15	
						2	10	2	
						3	5	1	
				14	2	1	6	9	
						2	9	8	
				15	2	1	7	15	
						2	8	3	
				16	2	1	5	2	
						2	9	14	
				17	3	1	10	3	
						2	6	13	
						3	8	5	
				18	2	1	5	4	
						2	9	10	
				19	4	1	6	7	
						2	8	2	
						3	5	11	
						4	10	6	
				20	3	1	9	5	
						2	10	14	
						3	6	1	

Table A.13 Data of problem P13

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	31	4	1	1	1	1	3	{(1, 2, 3), (1, 3, 4), (1, 4, 5)}
				2	3	1	10	5	
						2	7	4	
						3	5	10	
				3	4	1	6	8	
						2	8	2	
						3	7	5	

Table A.13 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	10	1	
				4	5	1	9	11	
						2	6	2	
						3	8	4	
						4	7	3	
						5	5	8	
2	F	68	10	1	1	1	1	13	{(2, 2, 3), (2, 3, 4), (2, 4, 2), (2, 5, 4), (2, 6, 5), (2, 7, 3), (2, 8, 5), (2, 9, 3), (2, 10, 2)}
				2	3	1	9	3	
						2	6	14	
						3	8	2	
				3	4	1	6	7	
						2	5	11	
						3	7	6	
						4	10	1	
				4	2	1	7	15	
						2	9	3	
				5	4	1	5	5	
						2	6	10	
						3	8	6	
						4	9	2	
				6	5	1	8	1	
						2	9	11	
						3	5	2	
						4	10	10	
						5	7	2	
				7	3	1	9	3	
						2	7	8	
						3	6	12	
				8	5	1	5	8	
						2	10	5	
						3	9	7	
						4	8	9	
						5	6	3	
				9	3	1	7	9	
						2	6	2	
						3	9	7	
				10	2	1	8	13	
						2	6	6	
3	F	96	11	1	1	1	3	12	{(3, 2, 3), (3, 3, 4), (3, 4, 5), (3, 5, 5), (3, 6, 3), (3, 7, 2), (3, 8, 2), (3, 9, 4), (3, 10, 3), (3, 11, 2)}
				2	3	1	5	15	
						2	7	4	
						3	9	1	
				3	4	1	5	5	
						2	10	3	
						3	6	14	

Table A.13 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	8	2	
				4	5	1	8	6	
						2	6	3	
						3	5	9	
						4	7	11	
						5	10	1	
				5	5	1	7	4	
						2	5	9	
						3	9	7	
						4	6	13	
						5	8	3	
				6	3	1	7	7	
						2	9	2	
						3	5	9	
				7	2	1	9	12	
						2	10	6	
				8	2	1	10	2	
						2	7	10	
				9	4	1	8	7	
						2	7	13	
						3	6	8	
						4	9	3	
				10	3	1	6	2	
						2	5	10	
						3	7	9	
				11	2	1	5	9	
						2	6	8	
4	T	41	5	1	2	1	2	9	{(4, 2, 1), (4, 3, 4)}
						2	5	4	
				2	1	1	3	6	{(4, 4, 3), (4, 5, 2)}
				3	4	1	9	8	
						2	6	4	
						3	10	13	
						4	7	3	
				4	3	1	9	5	
						2	7	14	
						3	6	1	
				5	2	1	8	9	
						2	5	3	
5	T	113	9	1	2	1	3	12	{(5, 2, 1), (5, 3, 2)}
						2	9	4	
				2	1	1	1	9	{(5, 4, 2), (5, 5, 3)}
				3	2	1	8	7	
						2	5	9	
				4	2	1	3	13	{(5, 6, 1), (5, 7, 5)}
						2	5	5	
				5	3	1	5	4	
						2	10	9	
						3	6	6	
				6	1	1	2	8	{(5, 8, 3), (5, 9, 3)}

Table A.13 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				7	5	1	10	9	
						2	6	5	
						3	8	8	
						4	5	6	
						5	9	4	
				8	3	1	10	2	
						2	7	11	
						3	5	7	
				9	3	1	7	3	
						2	8	7	
						3	6	10	
6	T	122	11	1	1	1	1	10	{(6, 2, 1), (6, 3, 3)}
				2	1	1	2	5	{(6, 4, 1), (6, 5, 3)}
				3	3	1	9	1	
						2	10	2	
						3	5	13	
				4	1	1	3	8	{(6, 6, 1), (6, 7, 5)}
				5	3	1	6	5	
						2	9	14	
						3	7	3	
				6	1	1	4	7	{(6, 8, 1), (6, 9, 2)}
				7	5	1	10	11	
						2	5	8	
						3	7	9	
						4	9	4	
						5	8	6	
				8	1	1	1	6	{(6, 10, 3), (6, 11, 4)}
				9	2	1	6	15	
						2	5	9	
				10	3	1	7	5	
						2	8	11	
						3	10	1	
				11	4	1	5	3	
						2	7	5	
						3	6	4	
						4	9	7	
7	C	57	27	1	1	1	4	12	{(7, 2, 1), (7, 3, 1), (7, 4, 1)}
				2	1	1	2	6	{(7, 5, 1), (7, 6, 1)}
				3	1	1	3	8	{(7, 7, 1), (7, 8, 1)}
				4	1	1	1	7	{(7, 9, 1), (7, 10, 1), (7, 11, 1)}
				5	1	1	2	5	{(7, 12, 2), (7, 13, 3)}
				6	1	1	4	8	{(7, 14, 3), (7, 15, 4), (7, 16, 2)}
				7	1	1	1	4	{(7, 17, 2), (7, 18, 3)}
				8	1	1	3	6	{(7, 19, 3), (7, 20, 4)}
				9	1	1	3	3	{(7, 21, 2), (7, 22, 2)}
				10	1	1	4	5	{(7, 23, 2), (7, 24, 3)}
				11	1	1	1	7	{(7, 25, 2), (7, 26, 3), (7, 27, 2)}
				12	2	1	6	10	
						2	8	6	
				13	3	1	10	8	

Table A.13 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	7	3	
						3	9	12	
				14	3	1	5	3	
						2	9	4	
						3	8	8	
				15	4	1	7	5	
						2	8	11	
						3	5	6	
						4	6	9	
				16	2	1	10	7	
						2	9	2	
				17	2	1	5	14	
						2	7	4	
				18	3	1	8	10	
						2	10	3	
						3	6	5	
				19	3	1	6	2	
						2	5	8	
						3	7	9	
				20	4	1	9	6	
						2	6	4	
						3	10	15	
						4	8	2	
				21	2	1	7	8	
						2	6	12	
				22	2	1	5	3	
						2	9	5	
				23	2	1	8	3	
						2	7	6	
				24	3	1	6	5	
						2	9	7	
						3	10	1	
				25	2	1	7	13	
						2	6	7	
				26	3	1	10	3	
						2	8	9	
						3	5	6	
				27	2	1	9	2	
						2	7	9	

Table A.14 Data of problem P14

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	34	4	1	2	1	2	7	{(1, 2, 2), (1, 3, 4), (1, 4, 3)}
						2	8	2	
				2	2	1	5	8	
						2	8	4	
				3	4	1	6	8	
						2	9	2	

Table A.14 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	10	12	
						4	7	3	
				4	3	1	8	9	
						2	7	6	
						3	5	5	
2	F	66	8	1	2	1	1	7	{(2, 2, 2), (2, 3, 4), (2, 4, 3), (2, 5, 5), (2, 6, 5), (2, 7, 3), (2, 8, 3)}
						2	5	5	
				2	2	1	5	2	
						2	8	6	
				3	4	1	9	1	
						2	10	7	
						3	6	9	
						4	7	3	
				4	3	1	6	2	
						2	5	3	
						3	7	12	
				5	5	1	10	2	
						2	9	8	
						3	8	6	
						4	7	7	
						5	6	9	
				6	5	1	6	3	
						2	5	12	
						3	7	6	
						4	8	3	
						5	9	1	
				7	3	1	10	15	
						2	6	2	
						3	9	1	
				8	3	1	5	4	
						2	9	7	
						3	7	8	
3	F	98	11	1	2	1	4	14	{(3, 2, 2), (3, 3, 4), (3, 4, 3), (3, 5, 3), (3, 6, 4), (3, 7, 2), (3, 8, 5), (3, 9, 4), (3, 10, 2), (3, 11, 2)}
						2	7	2	
				2	2	1	5	11	
						2	10	9	
				3	4	1	9	6	
						2	8	5	
						3	6	10	
						4	7	2	
				4	3	1	7	8	
						2	10	4	
						3	6	12	
				5	3	1	5	3	
						2	8	15	

Table A.14 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	9	9	
				6	4	1	10	4	
						2	9	15	
						3	7	2	
						4	8	1	
				7	2	1	10	14	
						2	6	1	
				8	5	1	5	6	
						2	6	9	
						3	9	1	
						4	7	14	
						5	10	3	
				9	4	1	9	8	
						2	8	10	
						3	7	1	
						4	5	13	
				10	2	1	8	3	
						2	5	8	
				11	2	1	6	9	
						2	7	2	
4	T	45	5	1	2	1	4	10	{(4, 2, 2), (4, 3, 4)}
						2	9	5	
				2	2	1	1	4	{(4, 4, 4), (4, 5, 3)}
						2	8	3	
				3	4	1	5	2	
						2	7	15	
						3	9	2	
						4	10	1	
				4	4	1	9	9	
						2	6	2	
						3	8	4	
						4	5	8	
				5	3	1	10	7	
						2	7	8	
						3	9	3	
5	T	101	9	1	1	1	4	12	{(5, 2, 2), (5, 3, 3)}
				2	2	1	2	5	{(5, 4, 2), (5, 5, 2)}
						2	7	7	
				3	3	1	5	4	
						2	6	9	
						3	8	1	
				4	2	1	4	9	{(5, 6, 2), (5, 7, 3)}
						2	6	2	
				5	2	1	10	10	
						2	9	5	
				6	2	1	1	4	{(5, 8, 3), (5, 9, 4)}
						2	9	3	
				7	3	1	6	15	
						2	9	6	
						3	5	1	

Table A.14 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				8	3	1	5	7	
						2	9	2	
						3	8	13	
				9	4	1	9	2	
						2	10	12	
						3	6	6	
						4	7	5	
6	T	122	11	1	1	1	2	9	{(6, 2, 2), (6, 3, 4)}
				2	2	1	4	11	{(6, 4, 1), (6, 5, 2)}
						2	9	8	
				3	4	1	8	6	
						2	7	12	
						3	10	5	
						4	5	9	
				4	1	1	1	10	{(6, 6, 2), (6, 7, 4)}
				5	2	1	7	9	
						2	6	4	
				6	2	1	3	6	{(6, 8, 1), (6, 9, 5)}
						2	5	11	
				7	4	1	10	2	
						2	6	7	
						3	9	3	
						4	5	10	
				8	1	1	2	5	{(6, 10, 3), (6, 11, 3)}
				9	5	1	5	4	
						2	9	5	
						3	8	3	
						4	6	11	
						5	10	3	
				10	3	1	9	7	
						2	10	5	
						3	7	4	
				11	3	1	6	9	
						2	8	1	
						3	9	2	
7	C	126	31	1	1	1	4	12	{(7, 2, 2), (7, 3, 2), (7, 4, 2)}
				2	2	1	1	9	{(7, 5, 1), (7, 6, 1), (7, 7, 1)}
						2	7	8	
				3	2	1	3	7	{(7, 8, 1), (7, 9, 1), (7, 10, 1)}
						2	9	5	
				4	2	1	2	10	{(7, 11, 1), (7, 12, 1), (7, 13, 1)}
						2	10	6	
				5	1	1	4	3	{(7, 14, 2), (7, 15, 5)}
				6	1	1	1	5	{(7, 16, 3), (7, 17, 4)}
				7	1	1	3	9	{(7, 18, 3), (7, 19, 3)}
				8	1	1	2	4	{(7, 20, 2), (7, 21, 2)}
				9	1	1	4	6	{(7, 22, 3), (7, 23, 4)}
				10	1	1	3	8	{(7, 24, 2), (7, 25, 3)}
				11	1	1	1	10	{(7, 26, 5), (7, 27, 2)}
				12	1	1	2	5	{(7, 28, 3), (7, 29, 4)}

Table A.14 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				13	1	1	3	6	{(7, 30, 2), (7, 31, 3)}
				14	2	1	9	14	
						2	5	1	
				15	5	1	6	4	
						2	5	10	
						3	10	6	
						4	7	5	
						5	8	2	
				16	3	1	10	2	
						2	7	15	
						3	9	3	
				17	4	1	8	11	
						2	5	9	
						3	10	1	
						4	6	8	
				18	3	1	7	10	
						2	8	3	
						3	10	5	
				19	3	1	5	9	
						2	6	5	
						3	9	6	
				20	2	1	10	6	
						2	6	11	
				21	2	1	9	4	
						2	7	9	
				22	3	1	7	5	
						2	6	13	
						3	10	2	
				23	4	1	8	3	
						2	9	4	
						3	10	1	
						4	5	7	
				24	2	1	10	6	
						2	8	8	
				25	3	1	6	2	
						2	7	10	
						3	5	3	
				26	5	1	9	1	
						2	7	8	
						3	6	3	
						4	10	4	
						5	8	9	
				27	2	1	10	14	
						2	5	2	
				28	3	1	9	12	
						2	8	3	
						3	6	1	
				29	4	1	5	6	
						2	10	1	
						3	7	8	

Table A.14 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	8	12	
				30	2	1	10	5	
						2	9	9	
				31	3	1	6	3	
						2	5	7	
						3	8	2	

Table A.15 Data of problem P15

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	64	7	1	1	1	1	5	{(1, 2, 3), (1, 3, 4), (1, 4, 3), (1, 5, 5), (1, 6, 4), (1, 7, 2)}
				2	3	1	8	5	
						2	5	8	
						3	6	3	
				3	4	1	6	8	
						2	10	7	
						3	9	6	
						4	5	2	
				4	3	1	9	5	
						2	7	2	
						3	8	9	
				5	5	1	6	2	
						2	7	13	
						3	8	1	
						4	5	4	
						5	10	7	
				6	4	1	5	3	
						2	6	7	
						3	10	6	
						4	9	2	
				7	2	1	7	11	
						2	5	5	
2	F	65	12	1	1	1	2	15	{(2, 2, 5), (2, 3, 4), (2, 4, 3), (2, 5, 4), (2, 6, 2), (2, 7, 2), (2, 8, 3), (2, 9, 4), (2, 10, 5), (2, 11, 3), (2, 12, 2)}
				2	5	1	9	7	
						2	7	2	
						3	10	14	
						4	6	3	
						5	8	2	
				3	4	1	7	7	
						2	6	4	
						3	9	1	
						4	5	13	
				4	3	1	6	2	
						2	5	9	
						3	9	1	

Table A.15 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				5	4	1	10	3	
						2	9	6	
						3	8	4	
						4	6	10	
				6	2	1	5	13	
						2	10	3	
				7	2	1	9	12	
						2	5	4	
				8	3	1	8	7	
						2	5	8	
						3	7	1	
				9	4	1	8	13	
						2	9	3	
						3	7	8	
						4	5	4	
				10	5	1	7	12	
						2	6	4	
						3	8	2	
						4	9	9	
						5	10	1	
				11	3	1	5	6	
						2	6	10	
						3	8	7	
				12	2	1	10	11	
						2	7	9	
3	T	56	7	1	1	1	3	8	{(3, 2, 1), (3, 3, 5)}
				2	1	1	2	5	{(3, 4, 1), (3, 5, 4)}
				3	5	1	10	7	
						2	8	3	
						3	6	11	
						4	7	4	
						5	9	1	
				4	1	1	4	3	{(3, 6, 3), (3, 7, 2)}
				5	4	1	9	3	
						2	5	12	
						3	6	1	
						4	8	8	
				6	3	1	5	2	
						2	8	9	
						3	9	4	
				7	2	1	7	15	
						2	10	6	
4	T	82	11	1	2	1	3	13	{(4, 2, 1), (4, 3, 3)}
						2	7	5	
				2	1	1	1	4	{(4, 4, 2), (4, 5, 5)}
				3	3	1	8	11	
						2	5	3	
						3	7	9	
				4	2	1	4	9	{(4, 6, 1), (4, 7, 3)}
						2	5	4	

Table A.15 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				5	5	1	10	7	
						2	9	5	
						3	7	2	
						4	8	6	
						5	6	10	
				6	1	1	2	12	{(4, 8, 2), (4, 9, 2)}
				7	3	1	5	1	
						2	8	13	
						3	6	5	
				8	2	1	3	8	{(4, 10, 4), (4, 11, 2)}
						2	10	3	
				9	2	1	9	5	
						2	7	4	
				10	4	1	6	1	
						2	9	2	
						3	8	6	
						4	10	15	
				11	2	1	7	7	
						2	5	14	
5	C	78	12	1	1	1	3	8	{(5, 2, 2), (5, 3, 2), (5, 4, 2)}
				2	2	1	4	7	{(5, 5, 5), (5, 6, 2)}
						2	10	3	
				3	2	1	1	10	{(5, 7, 3), (5, 8, 4), (5, 9, 2)}
						2	6	2	
				4	2	1	2	6	{(5, 10, 2), (5, 11, 5), (5, 12, 2)}
						2	8	5	
				5	5	1	6	6	
						2	5	4	
						3	9	11	
						4	10	6	
						5	8	7	
				6	2	1	7	10	
						2	5	3	
				7	3	1	10	8	
						2	8	3	
						3	6	12	
				8	4	1	5	10	
						2	6	1	
						3	8	13	
						4	10	2	
				9	2	1	6	5	
						2	9	1	
				10	2	1	10	11	
						2	7	5	
				11	5	1	8	15	
						2	7	3	
						3	10	8	
						4	9	2	
						5	6	4	
				12	2	1	9	8	

Table A.15 (Continued)

p	*PT*	D_p	n_p	*i*	J_{pi}	*j*	*m*	t_{pijm}	E_{pij}
						2	5	14	
6	C	96	14	1	2	1	1	9	{(6, 2, 1), (6, 3, 1), (6, 4, 1)}
						2	8	4	
				2	1	1	2	8	{(6, 5, 2), (6, 6, 5), (6, 7, 3)}
				3	1	1	3	10	{(6, 8, 3), (6, 9, 4), (6, 10, 2)}
				4	1	1	4	6	{(6, 11, 4), (6, 12, 3), (6, 13, 2), (6, 14, 2)}
				5	2	1	9	10	
						2	5	4	
				6	5	1	5	7	
						2	10	3	
						3	7	1	
						4	8	5	
						5	9	2	
				7	3	1	10	6	
						2	7	12	
						3	6	3	
				8	3	1	6	15	
						2	9	4	
						3	8	1	
				9	4	1	5	10	
						2	7	3	
						3	10	8	
						4	6	2	
				10	2	1	8	4	
						2	5	10	
				11	4	1	7	9	
						2	10	2	
						3	8	14	
						4	7	4	
				12	3	1	5	9	
						2	9	3	
						3	10	1	
				13	2	1	8	7	
						2	9	11	
				14	2	1	10	13	
							7	8	
7	C	59	22	1	1	1	1	8	{(7, 2, 2), (7, 3, 2)}
				2	2	1	2	11	{(7, 4, 1), (7, 5, 1), (7, 6, 1), (7, 7, 1)}
						2	10	4	
				3	2	1	4	5	{(7, 8, 1), (7, 9, 1)}
						2	6	3	
				4	1	1	3	3	{(7, 10, 2), (7, 11, 3)}
				5	1	1	4	5	{(7, 12, 4), (7, 13, 2)}
				6	1	1	2	5	{7, 14, 3), (7, 15, 3)}
				7	1	1	1	7	{(7, 16, 5), (7, 17, 2)}
				8	1	1	3	6	{(7, 18, 2), (7, 19, 4)}
				9	1	1	2	10	{(7, 20, 2), (7, 21, 2), (7, 22, 3)}
				10	2	1	5	7	

Table A.15 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	7	9	
				11	3	1	8	14	
						2	6	3	
						3	10	2	
				12	4	1	7	1	
						2	6	8	
						3	9	14	
						4	5	2	
				13	2	1	10	5	
						2	5	6	
				14	3	1	9	3	
						2	10	15	
						3	8	1	
				15	3	1	6	9	
						2	8	5	
						3	5	2	
				16	5	1	6	3	
						2	10	8	
						3	7	10	
						4	8	1	
						5	9	7	
				17	2	1	5	4	
						2	6	11	
				18	2	1	8	13	
						2	6	4	
				19	4	1	7	1	
						2	10	9	
						3	5	12	
						4	8	2	
				20	2	1	7	12	
						2	5	3	
				21	2	1	6	10	
						2	8	4	
				22	3	1	10	4	
						2	9	15	
						3	5	1	

Table A.16 Data of problem P16

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	44	7	1	2	1	4	9	{(1, 2, 4), (1, 3, 2), (1, 4, 4), (1, 5, 3), (1, 6, 2), (1, 7, 5)}
						2	5	11	
				2	4	1	6	1	
						2	5	12	
						3	7	2	
						4	9	3	
				3	2	1	8	13	
						2	6	1	

Table A.16 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				4	4	1	5	9	
						2	9	3	
						3	10	8	
						4	6	4	
				5	3	1	5	4	
						2	9	3	
						3	10	6	
				6	2	1	7	14	
						2	8	1	
				7	5	1	10	1	
						2	5	3	
						3	6	7	
						4	7	8	
						5	8	3	
2	F	122	12	1	2	1	4	14	{(2, 2, 2), (2, 3, 4), (2, 4, 5), (2, 5, 3), (2, 6, 5), (2, 7, 4), (2, 8, 2), (2, 9, 2), (2, 10, 3), (2, 11, 5), (2, 12, 2)}
						2	5	8	
				2	2	1	5	11	
						2	7	7	
				3	4	1	8	5	
						2	9	4	
						3	10	12	
						4	6	1	
				4	5	1	5	6	
						2	6	10	
						3	10	1	
						4	7	3	
						5	9	7	
				5	3	1	5	6	
						2	8	15	
						3	10	5	
				6	5	1	10	9	
						2	9	7	
						3	7	14	
						4	5	4	
						5	6	5	
				7	4	1	8	8	
						2	10	2	
						3	9	10	
						4	5	3	
				8	2	1	6	13	
						2	9	2	
				9	2	1	7	3	
						2	6	9	
				10	3	1	9	3	
						2	10	11	
						3	7	7	
				11	5	1	8	6	

Table A.16 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	9	2	
						3	5	10	
						4	7	1	
						5	10	2	
				12	2	1	9	9	
						2	5	4	
3	T	77	7	1	1	1	1	10	{(3, 2, 2), (3, 3, 3)}
				2	2	1	3	7	{(3, 4, 2), (3, 5, 2)}
						2	8	4	
				3	3	1	5	8	
						2	6	9	
						3	10	2	
				4	2	1	4	6	{(3, 6, 4), (3, 7, 3)}
						2	7	5	
				5	2	1	10	11	
						2	7	3	
				6	4	1	5	2	
						2	7	12	
						3	9	1	
						4	6	4	
				7	3	1	9	7	
						2	10	8	
						3	6	3	
4	T	76	11	1	1	1	4	7	{(4, 2, 2), (4, 3, 5)}
				2	2	1	3	5	{(4, 4, 2), (4, 5, 3)}
						2	7	3	
				3	5	1	5	5	
						2	7	15	
						3	9	5	
						4	6	4	
						5	8	11	
				4	2	1	2	10	{(4, 6, 2), (4, 7, 4)}
						2	10	6	
				5	3	1	9	4	
						2	8	2	
						3	5	12	
				6	2	1	1	9	{(4, 8, 2), (4, 9, 5)}
						2	5	7	
				7	4	1	8	10	
						2	7	1	
						3	9	3	
						4	5	3	
				8	2	1	4	6	{(4, 10, 2), (4, 11, 2)}
						2	6	3	
				9	5	1	7	14	
						2	5	9	
						3	10	1	
						4	9	2	
						5	8	3	
				10	2	1	8	1	

Table A.16 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	10	13	
				11	2	1	6	8	
						2	9	2	
5	C	96	13	1	2	1	4	12	{(5, 2, 1), (5, 3, 2), (5, 4, 1)}
						2	10	3	
				2	1	1	1	9	{(5, 5, 3), (5, 6, 4), (5, 7, 2), (5, 8, 3)}
				3	2	1	2	3	{(5, 9, 2), (5, 10, 5)}
						2	6	4	
				4	1	1	3	7	{(5, 11, 3), (5, 12, 2), (5, 13, 4)}
				5	3	1	9	13	
						2	6	4	
						3	10	1	
				6	4	1	5	4	
						2	6	3	
						3	8	2	
						4	7	12	
				7	2	1	8	9	
						2	10	4	
				8	3	1	7	11	
						2	6	2	
						3	5	6	
				9	2	1	6	15	
						2	9	5	
				10	5	1	10	1	
						2	9	7	
						3	8	3	
						4	5	10	
						5	7	5	
				11	3	1	9	3	
						2	8	14	
						3	5	4	
				12	2	1	10	10	
						2	7	8	
				13	4	1	8	5	
						2	6	2	
						3	9	11	
						4	5	6	
6	C	65	16	1	1	1	3	10	{(6, 2, 1), (6, 3, 1), (6, 4, 1), (6,5,1)}
				2	1	1	4	5	{(6, 6, 2), (6, 7, 3)}
				3	1	1	1	9	{(6, 8, 3), (6, 9, 2), (6, 10, 4)}
				4	1	1	2	7	{(6, 11, 2), (6, 12, 3), (6, 13, 3), (6, 14, 4)}
				5	1	1	3	4	{(6, 15, 5), (6, 16, 2)}
				6	2	1	9	14	
						2	5	3	
				7	3	1	10	3	
						2	8	1	
						3	7	10	
				8	3	1	7	2	

Table A.16 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	9	5	
						3	6	10	
				9	2	1	8	15	
						2	5	3	
				10	4	1	9	2	
						2	5	11	
						3	8	7	
						4	10	4	
				11	2	1	10	6	
							6	12	
				12	3	1	7	13	
							9	1	
							5	4	
				13	3	1	10	5	
						2	6	13	
						3	9	3	
				14	4	1	8	1	
						2	5	13	
						3	10	4	
						4	7	2	
				15	5	1	5	3	
						2	9	7	
						3	10	2	
						4	8	9	
						5	6	8	
				16	2	1	6	8	
						2	7	1	
7	C	59	20	1	2	1	2	11	{(7, 2, 1), (7, 3, 2)}
						2	5	4	
				2	1	1	3	10	{(7, 4, 1), (7, 5, 1), (7, 6, 1)}
				3	2	1	4	8	{(7, 7, 1), (7, 8, 1)}
						2	8	5	
				4	1	1	1	3	{(7, 9, 3), (7, 10, 4)}
				5	1	1	4	8	{(7, 11, 2), (7, 12, 5)}
				6	1	1	3	5	{(7, 13, 3), (7, 14, 2)}
				7	1	1	2	9	{(7, 15, 2), (7,16, 2), (7, 17, 3)}
				8	1	1	3	4	{(7, 18, 2), (7, 19, 4), (7, 20, 3)}
				9	3	1	5	3	
						2	7	6	
						3	8	4	
				10	4	1	6	8	
						2	10	2	
						3	5	2	
						4	9	10	
				11	2	1	10	2	
						2	6	12	
				12	5	1	9	1	
						2	5	10	
						3	6	3	
						4	7	8	

Table A.16 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						5	8	4	
				13	3	1	8	15	
						2	10	2	
						3	5	1	
				14	2	1	6	9	
						2	9	8	
				15	2	1	7	15	
						2	8	3	
				16	2	1	5	2	
						2	9	14	
				17	3	1	10	3	
						2	6	13	
						3	8	5	
				18	2	1	5	4	
						2	9	10	
				19	4	1	6	7	
						2	8	2	
						3	5	11	
						4	10	6	
				20	3	1	9	5	
						2	10	14	
						3	6	1	

Table A.17 Data of problem P17

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	35	6	1	1	1	5	12	{(1, 2, 5), (1, 3, 3), (1, 4, 2), (1, 5, 3), (1, 6, 4)}
				2	5	1	8	4	
						2	13	2	
						3	11	8	
						4	14	3	
						5	10	3	
				3	3	1	11	5	
						2	13	13	
						3	12	1	
				4	2	1	7	15	
						2	15	2	
				5	3	1	10	8	
						2	13	11	
						3	12	2	
				6	4	1	12	6	
						2	15	9	
						3	14	5	
						4	9	3	
2	T	60	5	1	1	1	1	5	{(2, 2, 1), (2, 3, 4)}
				2	1	1	2	7	{(2, 4, 5), (2, 5, 3)}
				3	4	1	7	11	
						2	11	3	

Table A.17 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	14	5	
						4	9	2	
				4	5	1	12	9	
						2	7	4	
						3	9	6	
						4	10	7	
						5	13	2	
				5	3	1	8	12	
						2	11	3	
						3	15	1	
3	T	82	5	1	2	1	5	9	{(3, 2, 1), (3, 3, 4)}
						2	9	4	
				2	1	1	6	6	{(3, 4, 3), (3, 5, 2)}
				3	4	1	13	8	
						2	8	4	
						3	14	13	
						4	10	3	
				4	3	1	8	5	
						2	7	14	
						3	15	1	
				5	2	1	11	9	
						2	9	3	
4	T	52	7	1	2	1	3	8	{(4, 2, 2), (4, 3, 3)}
						2	11	2	
				2	2	1	5	7	{(4, 4, 1), (4, 5, 3)}
						2	7	3	
				3	3	1	7	3	
						2	11	5	
						3	9	7	
				4	1	1	1	5	{(4, 6, 2), (4, 7, 4)}
				5	3	1	8	4	
						2	13	13	
						3	10	1	
				6	2	1	10	9	
						2	7	5	
				7	4	1	14	1	
						2	15	7	
						3	9	15	
						4	11	4	
5	C	75	15	1	1	1	2	13	{(5, 2, 1), (5, 3, 1), (5, 4, 1), (5, 5, 1)}
				2	1	1	3	6	{(5, 6, 5), (5, 7, 3)}
				3	1	1	1	4	{(5, 8, 2), (5, 9, 3)}
				4	1	1	4	8	{(5, 10, 5), (5, 11, 2), (5, 12, 3)}
				5	1	1	5	9	{(5, 13, 4), (5, 14, 2), (5, 15, 2)}
				6	5	1	7	1	
						2	15	3	
						3	11	8	
						4	10	15	
						5	12	2	

Table A.17 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				7	3	1	9	4	
						2	8	12	
						3	13	5	
				8	2	1	14	4	
						2	7	13	
				9	3	1	10	7	
						2	12	5	
						3	9	11	
				10	5	1	15	11	
						2	8	4	
						3	7	5	
						4	10	8	
						5	11	1	
				11	2	1	14	2	
						2	12	10	
				12	3	1	8	4	
						2	9	5	
						3	13	4	
				13	4	1	12	4	
						2	7	6	
						3	10	4	
						4	15	3	
				14	2	1	13	10	
						2	14	9	
				15	2	1	11	8	
						2	9	3	

Table A.18 Data of problem P18

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	34	6	1	2	1	6	7	{(1, 2, 2), (1, 3, 4), (1, 4, 3), (1, 5, 5), (1, 6, 2)}
						2	9	4	
				2	2	1	11	15	
						2	8	2	
				3	4	1	13	2	
						2	9	9	
						3	7	6	
						4	14	3	
				4	3	1	9	1	
						2	13	10	
						3	10	5	
				5	5	1	15	4	
						2	7	3	
						3	11	7	
						4	9	8	
						5	12	1	
				6	2	1	8	11	
						2	7	3	

Table A.18 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
2	T	56	5	1	1	1	3	8	{(2, 2, 2), (2, 3, 5)}
				2	2	1	4	8	{(2, 4, 2), (2, 5, 3)}
						2	8	3	
				3	5	1	15	2	
						2	14	9	
						3	7	4	
						4	12	6	
						5	10	1	
				4	2	1	9	12	
						2	13	5	
				5	3	1	11	8	
						2	8	7	
						3	14	3	
3	T	90	5	1	2	1	1	10	{(3, 2, 2), (3, 3, 4)}
						2	7	5	
				2	2	1	2	4	{(3, 4, 4), (3, 5, 3)}
						2	15	3	
				3	4	1	10	2	
						2	9	15	
						3	11	2	
						4	8	1	
				4	4	1	8	9	
						2	7	2	
						3	10	4	
						4	11	8	
				5	3	1	13	7	
						2	15	8	
						3	7	3	
4	T	64	9	1	1	1	5	11	{(4, 2, 2), (4, 3, 5)}
				2	2	1	6	8	{(4, 4, 1), (4, 5, 3)}
						2	14	3	
				3	5	1	15	1	
						2	7	9	
						3	13	5	
						4	9	4	
						5	10	8	
				4	1	1	1	6	{(4, 6, 2), (4, 7, 2)}
				5	3	1	14	7	
						2	10	11	
						3	11	2	
				6	2	1	2	7	{(4, 8, 4), (4, 9, 2)}
						2	9	6	
				7	2	1	11	10	
						2	8	5	
				8	4	1	9	5	
						2	10	13	
						3	11	2	
						4	7	3	
				9	2	1	14	8	
						2	8	5	

Table A.18 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
5	C	72	16	1	1	1	2	13	{(5, 2, 2), (5, 3, 2), (5, 4, 2), (5, 5, 2)}
				2	2	1	3	10	{(5, 6, 2), (5, 7, 3), (5, 8, 2)}
						2	15	4	
				3	2	1	4	5	{(5, 9, 3), (5, 10, 5)}
						2	9	5	
				4	2	1	5	4	{(5, 11, 4), (5, 12, 3)}
						2	7	6	
				5	2	1	6	11	{(5, 13, 2), (5, 14, 3), (5, 15, 2), (5, 16, 3)}
						2	13	4	
				6	2	1	10	9	
						2	12	10	
				7	3	1	15	11	
						2	13	9	
						3	9	1	
				8	2	1	8	2	
						2	7	8	
				9	3	1	9	14	
						2	15	3	
						3	13	6	
				10	5	1	7	5	
						2	8	7	
						3	12	4	
						4	14	8	
						5	11	1	
				11	4	1	13	1	
						2	12	3	
						3	14	15	
						4	10	4	
				12	3	1	8	7	
						2	9	8	
						3	11	4	
				13	2	1	13	5	
						2	10	7	
				14	3	1	12	5	
						2	7	13	
						3	14	2	
				15	2	1	14	1	
						2	8	12	
				16	3	1	11	10	
						2	9	4	
						3	15	6	

Table A.19 Data of problem P19

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	62	4	1	1	1	1	3	{(1, 2, 3), (1, 3, 4), (1, 4, 5)}
				2	3	1	11	5	

Table A.19 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	8	4	
						3	12	10	
				3	4	1	12	8	
						2	7	2	
						3	15	5	
						4	10	1	
				4	5	1	13	11	
						2	9	2	
						3	15	4	
						4	11	3	
						5	14	8	
2	F	48	7	1	1	1	3	5	{(2, 2, 3), (2, 3, 4), (2, 4, 3), (2, 5, 5), (2, 6, 4), (2, 7, 2)}
				2	3	1	12	5	
						2	15	8	
						3	7	3	
				3	4	1	13	8	
						2	9	7	
						3	11	6	
						4	15	2	
				4	3	1	10	5	
						2	14	2	
						3	9	9	
				5	5	1	8	2	
						2	9	13	
						3	7	1	
						4	10	4	
						5	12	7	
				6	4	1	11	3	
						2	7	7	
						3	9	6	
						4	13	2	
				7	2	1	15	11	
						2	8	5	
3	T	37	7	1	1	1	3	8	{(3, 2, 1), (3, 3, 5)}
				2	1	1	4	5	{(3, 4, 1), (3, 5, 4)}
				3	5	1	15	7	
						2	8	3	
						3	7	11	
						4	13	4	
						5	10	1	
				4	1	1	5	3	{(3, 6, 3), (3, 7, 2)}
				5	4	1	13	3	
						2	11	12	
						3	8	1	
						4	12	8	
				6	3	1	7	2	
						2	8	9	
						3	14	4	
				7	2	1	10	15	

Table A.19 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	9	6	
4	T	60	9	1	1	1	1	9	{(4, 2, 1), (4, 3, 4)}
				2	1	1	2	4	{(4, 4, 1), (4, 5, 2)}
				3	4	1	11	15	
						2	7	2	
						3	8	3	
						4	10	7	
				4	1	1	3	5	{(4, 6, 1), (4, 7, 3)}
				5	2	1	12	9	
						2	9	6	
				6	1	1	4	7	{(4, 8, 2), (4, 9, 3)}
				7	3	1	14	4	
						2	8	14	
						3	12	1	
				8	2	1	7	13	
						2	13	2	
				9	3	1	10	3	
						2	11	1	
						3	15	11	
5	C	118	22	1	1	1	6	9	{(5, 2, 1), (5, 3, 1)}
				2	1	1	3	5	{(5, 4, 1), (5, 5, 1), (5, 6, 1)}
				3	1	1	5	8	{(5, 7, 1), (5, 8, 1), (5, 9, 1)}
				4	1	1	1	6	{(5, 10, 3), (5, 11, 3)}
				5	1	1	2	10	{(5, 12, 4), (5, 13, 2)}
				6	1	1	5	3	{(5, 14, 3), (5, 15, 2)}
				7	1	1	4	6	{(5, 16, 3), (5, 17, 2), (5, 18, 3)}
				8	1	1	3	5	{(5, 19, 5), (5, 20, 2)}
				9	1	1	6	7	{(5, 21, 2), (5, 22, 2)}
				10	3	1	15	12	
						2	10	1	
						3	8	7	
				11	3	1	9	11	
						2	12	1	
						3	13	3	
				12	4	1	7	13	
						2	14	5	
						3	9	2	
						4	11	3	
				13	2	1	8	12	
						2	10	4	
				14	3	1	13	8	
						2	11	14	
						3	9	5	
				15	2	1	10	10	
						2	7	3	
				16	3	1	11	8	
						2	9	4	
						3	12	13	
				17	2	1	10	11	
						2	14	5	

Table A.19 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				18	3	1	7	6	
						2	13	9	
						3	15	8	
				19	5	1	12	4	
						2	13	15	
						3	11	3	
						4	15	8	
						5	8	7	
				20	2	1	7	5	
						2	9	10	
				21	2	1	10	9	
						2	11	3	
				22	2	1	14	13	
						2	7	2	

Table A.20 Data of problem P20

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	68	4	1	2	1	2	7	{(1, 2, 2), (1, 3, 4), (1, 4, 3)}
						2	12	2	
				2	2	1	7	8	
						2	10	4	
				3	4	1	12	8	
						2	13	2	
						3	8	12	
						4	15	3	
				4	3	1	9	9	
						2	11	6	
						3	14	5	
2	F	66	7	1	2	1	4	9	{(2, 2, 4), (2, 3, 2), (2, 4, 4), (2, 5, 3), (2, 6, 2), (2, 7, 5)}
						2	8	11	
				2	4	1	9	1	
						2	13	12	
						3	11	2	
						4	12	3	
				3	2	1	15	13	
						2	13	1	
				4	4	1	10	9	
						2	7	3	
						3	10	8	
						4	9	4	
				5	3	1	8	4	
						2	14	3	
						3	11	6	
				6	2	1	12	14	
						2	7	1	
				7	5	1	14	1	
						2	8	3	

Table A.20 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	11	7	
						4	7	8	
						5	10	3	
3	T	51	7	1	1	1	6	10	{(3, 2, 2), (3, 3, 3)}
				2	2	1	1	7	{(3, 4, 2), (3, 5, 2)}
						2	12	4	
				3	3	1	15	8	
						2	10	9	
						3	8	2	
				4	2	1	2	6	{(3, 6, 4), (3, 7, 3)}
						2	15	5	
				5	2	1	14	11	
						2	7	3	
				6	4	1	10	2	
						2	14	12	
						3	11	1	
						4	8	4	
				7	3	1	13	7	
						2	7	8	
						3	9	3	
4	T	101	9	1	1	1	2	12	{(4, 2, 2), (4, 3, 3)}
				2	2	1	4	5	{(4, 4, 2), (4, 5, 2)}
						2	14	7	
				3	3	1	11	4	
						2	7	9	
						3	12	1	
				4	2	1	6	9	{(4, 6, 2), (4, 7, 3)}
						2	8	2	
				5	2	1	13	10	
						2	15	5	
				6	2	1	3	4	{(4, 8, 3), (4, 9, 4)}
						2	13	3	
				7	3	1	10	15	
						2	8	6	
						3	11	1	
				8	3	1	10	7	
						2	9	2	
						3	14	13	
				9	4	1	13	2	
						2	7	12	
						3	12	6	
						4	13	5	
5	C	106	17	1	2	1	4	10	{(5, 2, 1), (5, 3, 2), (5, 4, 2), (5, 5, 2)}
						2	8	5	
				2	1	1	5	8	{(5, 6, 2), (5, 7, 3), (5, 8, 3)}
				3	2	1	6	6	{(5, 9, 3), (5, 10, 4), (5, 11, 2)}
						2	7	3	
				4	2	1	1	11	{(5, 12, 2), (5, 13, 2), (5, 14, 3), (5, 15, 4)}

Table A.20 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	9	2	
				5	2	1	2	3	{(5, 16, 5), (5, 17, 2)}
						2	14	4	
				6	2	1	15	10	
						2	12	2	
				7	3	1	10	3	
						2	7	15	
						3	9	1	
				8	3	1	11	8	
						2	8	2	
						3	14	5	
				9	3	1	7	1	
						2	15	3	
						3	10	12	
				10	4	1	13	12	
						2	14	4	
						3	12	5	
						4	11	1	
				11	2	1	9	14	
						2	8	6	
				12	2	1	8	14	
						2	7	4	
				13	2	1	12	4	
						2	15	13	
				14	3	1	14	10	
						2	10	8	
						3	13	3	
				15	4	1	15	6	
						2	9	4	
						3	12	11	
						4	11	1	
				16	5	1	7	2	
						2	15	7	
						3	8	3	
						4	11	10	
						5	12	9	
				17	2	1	10	6	
						2	9	8	

Table A.21 Data of problem P21

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	48	5	1	1	1	3	8	{(1, 2, 3), (1, 3, 4), (1, 4, 3), (1, 5, 5)}
				2	3	1	11	2	
						2	9	13	
						3	10	6	
				3	4	1	13	4	
						2	9	1	

Table A.21 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	15	3	
						4	12	9	
				4	3	1	7	5	
						2	13	3	
						3	8	11	
				5	5	1	12	2	
						2	10	8	
						3	14	10	
						4	7	1	
						5	11	3	
2	T	40	9	1	1	1	1	9	{(2, 2, 1), (2, 3, 4)}
				2	1	1	2	4	{(2, 4, 1), (2, 5, 2)}
				3	4	1	11	15	
						2	7	2	
						3	8	3	
						4	10	7	
				4	1	1	3	5	{(2, 6, 1), (2, 7, 3)}
				5	2	1	12	9	
						2	9	6	
				6	1	1	4	7	{(2, 8, 2), (2, 9, 3)}
				7	3	1	14	4	
						2	8	14	
						3	12	1	
				8	2	1	7	13	
						2	13	2	
				9	3	1	10	3	
						2	11	1	
						3	15	11	
3	C	88	12	1	1	1	1	9	{(3, 2, 1), (3, 3, 1), (3, 4, 1)}
				2	1	1	2	7	{(3, 5, 3), (3, 6, 2), (3, 7, 4)}
				3	1	1	3	5	{(3, 8, 3), (3, 9, 5)}
				4	1	1	4	10	{(3, 10, 2), (3, 11, 3), (3, 12, 2)}
				5	3	1	9	11	
						2	10	2	
						3	13	4	
				6	2	1	12	5	
						2	7	8	
				7	4	1	15	3	
						2	11	13	
						3	14	1	
						4	8	5	
				8	3	1	14	6	
						2	15	7	
						3	10	12	
				9	5	1	7	8	
						2	10	7	
						3	9	2	
						4	11	3	
						5	13	10	
				10	2	1	8	15	

Table A.21 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	12	3	
				11	3	1	7	4	
						2	15	14	
						3	9	3	
				12	2	1	10	6	
						2	14	9	
4	C	78	12	1	1	1	3	8	{(4, 2, 2), (4, 3, 2), (4, 4, 2)}
				2	2	1	5	7	{(4, 5, 5), (4, 6, 2)}
						2	9	3	
				3	2	1	6	10	{(4, 7, 3), (4, 8, 4), (4, 9, 2)}
						2	12	2	
				4	2	1	2	6	{(4, 10, 2), (4, 11, 5), (4, 12, 2)}
						2	7	5	
				5	5	1	11	6	
						2	9	4	
						3	15	11	
						4	7	6	
						5	14	7	
				6	2	1	12	10	
						2	10	3	
				7	3	1	7	8	
						2	14	3	
						3	13	12	
				8	4	1	8	10	
						2	12	1	
						3	9	13	
						4	11	2	
				9	2	1	15	5	
						2	10	1	
				10	2	1	14	11	
						2	12	5	
				11	5	1	10	15	
						2	13	3	
						3	15	8	
						4	11	2	
						5	8	4	
				12	2	1	9	8	
						2	7	14	
5	C	48	14	1	2	1	4	9	{(5, 2, 1), (5, 3, 1), (5, 4, 1)}
						2	10	4	
				2	1	1	6	8	{(5, 5, 2), (5, 6, 5), (5, 7, 3)}
				3	1	1	5	10	{(5, 8, 3), (5, 9, 4), (5, 10, 2)}
				4	1	1	1	6	{(5, 11, 4), (5, 12, 3), (5, 13, 2), (5, 14, 2)}
				5	2	1	7	10	
						2	8	4	
				6	5	1	14	7	
						2	7	3	
						3	9	1	
						4	13	5	

Table A.21 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						5	12	2	
				7	3	1	15	6	
						2	10	12	
						3	11	3	
				8	3	1	8	15	
						2	14	4	
						3	10	1	
				9	4	1	12	10	
						2	11	3	
						3	12	8	
						4	7	2	
				10	2	1	9	4	
						2	15	10	
				11	4	1	11	9	
						2	7	2	
						3	9	14	
						4	8	4	
				12	3	1	13	9	
						2	12	3	
						3	9	1	
				13	2	1	10	7	
						2	15	11	
				14	2	1	14	13	
							13	8	

Table A.22 Data of problem P22

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	68	5	1	2	1	4	7	{(1, 2, 4), (1, 3, 2), (1, 4, 4), (1, 5, 3)}
						2	15	3	
				2	4	1	11	14	
						2	8	7	
						3	7	6	
						4	13	8	
				3	2	1	13	15	
						2	10	4	
				4	4	1	7	2	
						2	15	5	
						3	12	13	
						4	8	4	
				5	3	1	14	9	
						2	9	8	
						3	11	2	
2	T	56	7	1	2	1	2	4	{(2, 2, 1), (2, 3, 4)}
						2	12	3	
				2	1	1	4	9	{(2, 4, 2), (2, 5, 5)}
				3	4	1	14	4	
						2	9	3	

Table A.22 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	7	12	
						4	10	8	
				4	2	1	6	11	{(2, 6, 3), (2, 7, 3)}
						2	9	7	
				5	5	1	15	5	
						2	10	3	
						3	13	8	
						4	7	10	
						5	11	6	
				6	3	1	8	8	
						2	13	7	
						3	11	4	
				7	3	1	10	14	
						2	7	6	
						3	15	2	
3	C	96	12	1	1	1	5	12	{(3, 2, 1), (3, 3, 1), (3, 4, 1)}
				2	1	1	6	8	{(3, 5, 3), (3, 6, 4)}
				3	1	1	1	6	{(3, 7, 3), (3, 8, 3)}
				4	1	1	2	7	{(3, 9, 2), (3, 10, 2), (3, 11, 4), (3, 12, 5)}
				5	3	1	8	3	
						2	11	8	
						3	13	7	
				6	4	1	9	1	
						2	10	6	
						3	7	10	
						4	14	9	
				7	3	1	12	2	
						2	9	13	
						3	7	4	
				8	3	1	15	3	
						2	8	11	
						3	10	5	
				9	2	1	13	10	
						2	14	1	
				10	2	1	7	4	
						2	15	12	
				11	4	1	11	3	
						2	12	14	
						3	13	1	
						4	9	4	
				12	5	1	14	5	
						2	8	4	
						3	9	2	
						4	10	15	
						5	7	3	
4	C	72	13	1	2	1	1	12	{(4, 2, 1), (4, 3, 2), (4, 4, 1)}
						2	13	3	
				2	1	1	4	9	{(4, 5, 3), (4, 6, 4), (4, 7, 2), (4, 8, 3)}

Table A.22 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				3	2	1	5	3	{(4, 9, 2), (4, 10, 5)}
						2	7	4	
				4	1	1	2	7	{(4, 11, 3), (4, 12, 2), (4, 13, 4)}
				5	3	1	7	13	
						2	14	4	
						3	11	1	
				6	4	1	8	4	
						2	12	3	
						3	9	2	
						4	10	12	
				7	2	1	14	9	
						2	9	4	
				8	3	1	13	11	
						2	8	2	
						3	15	6	
				9	2	1	9	15	
						2	8	5	
				10	5	1	15	1	
						2	10	7	
						3	14	3	
						4	12	10	
						5	7	5	
				11	3	1	9	3	
						2	8	14	
						3	13	4	
				12	2	1	11	10	
						2	15	8	
				13	4	1	10	5	
						2	7	2	
						3	14	11	
						4	12	6	
5	C	59	20	1	2	1	6	11	{(5, 2, 1), (5, 3, 2)}
						2	13	4	
				2	1	1	1	10	{(5, 4, 1), (5, 5, 1), (5, 6, 1)}
				3	2	1	2	8	{(5, 7, 1), (5, 8, 1)}
						2	7	5	
				4	1	1	3	3	{(5, 9, 3), (5, 10, 4)}
				5	1	1	4	8	{(5, 11, 2), (5, 12, 5)}
				6	1	1	5	5	{(5, 13, 3), (5, 14, 2)}
				7	1	1	6	9	{(5, 15, 2), (5,16, 2), (5, 17, 3)}
				8	1	1	1	4	{(5, 18, 2), (5, 19, 4), (5, 20, 3)}
				9	3	1	7	3	
						2	10	6	
						3	8	4	
				10	4	1	15	8	
						2	14	2	
						3	11	2	
						4	13	10	
				11	2	1	10	2	
						2	7	12	

Table A.22 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				12	5	1	8	1	
						2	12	10	
						3	14	3	
						4	11	8	
						5	15	4	
				13	3	1	9	15	
						2	13	2	
						3	12	1	
				14	2	1	11	9	
						2	14	8	
				15	2	1	13	15	
						2	9	3	
				16	2	1	14	2	
						2	8	14	
				17	3	1	7	3	
						2	10	13	
						3	11	5	
				18	2	1	9	4	
						2	15	10	
				19	4	1	10	7	
						2	7	2	
						3	14	11	
						4	12	6	
				20	3	1	8	5	
						2	11	14	
						3	13	1	

Table A.23 Data of problem P23

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	31	4	1	1	1	1	3	{(1, 2, 3), (1, 3, 4), (1, 4, 5)}
				2	3	1	11	5	
						2	8	4	
						3	12	10	
				3	4	1	12	8	
						2	7	2	
						3	15	5	
						4	10	1	
				4	5	1	13	11	
						2	9	2	
						3	15	4	
						4	11	3	
						5	14	8	
2	F	53	9	1	1	1	5	9	{(2, 2, 5), (2, 3, 4), (2, 4, 2), (2, 5, 3), (2, 6, 4), (2, 7, 5), (2, 8, 2), (2, 9, 3)}
				2	5	1	10	3	
						2	7	2	
						3	11	6	

Table A.23 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	13	8	
						5	14	3	
				3	4	1	14	1	
						2	8	4	
						3	9	2	
						4	10	10	
				4	2	1	9	11	
						2	7	9	
				5	3	1	13	2	
						2	8	7	
						3	12	5	
				6	4	1	7	2	
						2	9	4	
						3	15	13	
						4	8	1	
				7	5	1	8	2	
						2	14	7	
						3	11	3	
						4	7	6	
						5	10	8	
				8	2	1	15	1	
						2	9	15	
				9	3	1	7	9	
						2	12	7	
						3	11	5	
3	F	76	13	1	1	1	3	11	{(3, 2, 3), (3, 3, 4), (3, 4, 3), (3, 5, 2), (3, 6, 2), (3, 7, 3), (3, 8, 5), (3, 9, 3), (3, 10, 4), (3, 11, 2), (3, 12, 2), (3, 13, 5)}
				2	3	1	9	10	
						2	13	7	
						3	10	8	
				3	4	1	11	3	
						2	14	6	
						3	10	4	
						4	9	13	
				4	3	1	8	2	
						2	7	12	
						3	14	8	
				5	2	1	15	13	
						2	9	2	
				6	2	1	8	12	
						2	12	3	
				7	3	1	10	1	
						2	11	3	
						3	14	14	
				8	5	1	13	5	
						2	10	6	
						3	12	1	
						4	15	9	

Table A.23 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						5	8	3	
				9	3	1	14	3	
						2	8	15	
						3	7	6	
				10	4	1	12	3	
						2	11	8	
						3	10	9	
						4	14	7	
				11	2	1	9	2	
						2	15	11	
				12	2	1	14	8	
						2	8	9	
				13	5	1	7	2	
						2	15	5	
						3	14	4	
						4	9	9	
						5	13	1	
4	T	37	7	1	1	1	3	8	{(4, 2, 1), (4, 3, 5)}
				2	1	1	4	5	{(4, 4, 1), (4, 5, 4)}
				3	5	1	15	7	
						2	8	3	
						3	7	11	
						4	13	4	
						5	10	1	
				4	1	1	5	3	{(4, 6, 3), (4, 7, 2)}
				5	4	1	13	3	
						2	11	12	
						3	8	1	
						4	12	8	
				6	3	1	7	2	
						2	8	9	
						3	14	4	
				7	2	1	10	15	
						2	9	6	
5	T	78	7	1	2	1	3	8	{(5, 2, 2), (5, 3, 3)}
						2	11	2	
				2	2	1	5	7	{(5, 4, 1), (5, 5, 3)}
						2	7	3	
				3	3	1	7	3	
						2	11	5	
						3	9	7	
				4	1	1	1	5	{(5, 6, 2), (5, 7, 4)}
				5	3	1	8	4	
						2	13	13	
						3	10	1	
				6	2	1	10	9	
						2	7	5	
				7	4	1	14	1	
						2	15	7	
						3	9	15	

Table A.23 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	11	4	
6	T	150	9	1	2	1	3	12	{(6, 2, 1), (6, 3, 2)}
						2	15	4	
				2	1	1	4	9	{(6, 4, 2), (6, 5, 3)}
				3	2	1	11	7	
						2	8	9	
				4	2	1	5	13	{(6, 6, 1), (6, 7, 5)}
						2	7	5	
				5	3	1	7	4	
						2	10	9	
						3	13	6	
				6	1	1	6	8	{(6, 8, 3), (6, 9, 3)}
				7	5	1	15	9	
						2	10	5	
						3	12	8	
						4	8	6	
						5	14	4	
				8	3	1	8	2	
						2	14	11	
						3	10	7	
				9	3	1	9	3	
						2	11	7	
						3	13	10	
7	C	61	26	1	2	1	3	10	{(7, 2, 1), (7, 3, 1), (7, 4, 2)}
						2	8	5	
				2	1	1	4	9	{(7, 5, 1), (7, 6, 1)}
				3	1	1	5	6	{(7, 7, 2), (7, 8, 2)}
				4	2	1	6	5	{(7, 9, 1), (7, 10, 1), (7, 11, 1)}
						2	14	4	
				5	1	1	1	7	{(7, 12, 2), (7, 13, 3), (7, 14, 3)}
				6	1	1	2	3	{(7, 15, 2), (7, 16, 2)}
				7	2	1	3	5	{(7, 17, 3), (7, 18, 4)}
						2	9	2	
				8	2	1	6	4	{(7, 19, 2), (7, 20, 3)}
						2	13	5	
				9	1	1	2	3	{(7, 21, 3), (7, 22, 3)}
				10	1	1	4	5	{(7, 23, 5), (7, 24, 2)}
				11	1	1	5	6	{(7, 25, 2), (7, 26, 3)}
				12	2	1	9	4	
						2	13	10	
				13	3	1	7	1	
						2	12	11	
						3	15	2	
				14	3	1	14	2	
						2	10	12	
						3	11	5	
				15	2	1	8	15	
						2	14	3	
				16	2	1	11	10	
						2	9	1	

Table A.23 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				17	3	1	15	2	
						2	11	10	
						3	8	3	
				18	4	1	13	11	
						2	7	1	
						3	12	4	
						4	10	2	
				19	2	1	10	5	
						2	7	10	
				20	3	1	12	1	
						2	9	6	
						3	14	9	
				21	3	1	14	8	
						2	10	5	
						3	9	14	
				22	3	1	15	13	
						2	8	4	
						3	12	7	
				23	5	1	9	2	
						2	8	7	
						3	12	9	
						4	14	2	
						5	7	12	
				24	2	1	11	3	
						2	13	6	
				25	2	1	10	9	
						2	15	7	
				26	3	1	7	6	
						2	14	8	
						3	11	3	

Table A.24 Data of problem P24

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	34	4	1	2	1	2	7	{(1, 2, 2), (1, 3, 4), (1, 4, 3)}
						2	12	2	
				2	2	1	7	8	
						2	10	4	
				3	4	1	12	8	
						2	13	2	
						3	8	12	
						4	15	3	
				4	3	1	9	9	
						2	11	6	
						3	14	5	
2	F	66	9	1	2	1	6	8	{(2, 2, 2), (2, 3, 5), (2, 4, 3), (2, 5, 3), (2, 6, 4), (2, 7, 2), (2, 8, 4), (2, 9, 5)}
						2	14	7	

Table A.24 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				2	2	1	8	13	
						2	10	6	
				3	5	1	12	6	
						2	9	3	
						3	3	10	
						4	15	5	
						5	14	3	
				4	3	1	7	13	
						2	10	8	
						3	9	6	
				5	3	1	14	11	
						2	12	7	
						3	11	4	
				6	4	1	9	2	
						2	10	13	
						3	15	1	
						4	12	5	
				7	2	1	13	15	
						2	8	3	
				8	4	1	11	7	
						2	15	14	
						3	10	5	
						4	13	1	
				9	5	1	10	3	
						2	12	9	
						3	13	4	
						4	9	8	
						5	7	5	
3	F	88	13	1	2	1	4	13	{(3, 2, 5), (3, 3, 2), (3, 4, 3), (3, 5, 3), (3, 6, 5), (3, 7, 2), (3, 8, 3), (3, 9, 2), (3, 10, 5), (3, 11, 3), (3, 12, 4), (3, 13, 4)}
						2	7	3	
				2	5	1	10	1	
						2	7	11	
						3	14	6	
						4	15	3	
						5	11	7	
				3	2	1	7	9	
						2	13	8	
				4	3	1	15	4	
						2	9	15	
						3	10	8	
				5	3	1	7	3	
						2	10	2	
						3	15	10	
				6	5	1	11	5	
						2	9	3	
						3	14	4	
						4	12	13	

Table A.24 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						5	8	1	
				7	2	1	9	3	
						2	15	12	
				8	3	1	12	9	
						2	8	4	
						3	7	8	
				9	2	1	14	10	
						2	10	2	
				10	5	1	7	2	
						2	9	5	
						3	13	3	
						4	11	13	
						5	8	1	
				11	3	1	9	11	
						2	14	1	
						3	7	4	
				12	4	1	8	2	
						2	15	7	
						3	11	6	
						4	9	12	
				13	4	1	13	4	
						2	7	6	
						3	10	5	
						4	12	7	
4	T	51	7	1	1	1	6	10	{(4, 2, 2), (4, 3, 3)}
				2	2	1	1	7	{(4, 4, 2), (4, 5, 2)}
						2	12	4	
				3	3	1	15	8	
						2	10	9	
						3	8	2	
				4	2	1	2	6	{(4, 6, 4), (4, 7, 3)}
						2	15	5	
				5	2	1	14	11	
						2	7	3	
				6	4	1	10	2	
						2	14	12	
						3	11	1	
						4	8	4	
				7	3	1	13	7	
						2	7	8	
						3	9	3	
5	T	84	7	1	2	1	2	4	{(5, 2, 1), (5, 3, 4)}
						2	12	3	
				2	1	1	4	9	{(5, 4, 2), (5, 5, 5)}
				3	4	1	14	4	
						2	9	3	
						3	7	12	
						4	10	8	
				4	2	1	6	11	{(5, 6, 3), (5, 7, 3)}
						2	9	7	

Table A.24 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				5	5	1	15	5	
						2	10	3	
						3	13	8	
						4	7	10	
						5	11	6	
				6	3	1	8	8	
						2	13	7	
						3	11	4	
				7	3	1	10	14	
						2	7	6	
						3	15	2	
6	T	134	9	1	1	1	2	12	{(6, 2, 2), (6, 3, 3)}
				2	2	1	4	5	{(6, 4, 2), (6, 5, 2)}
						2	14	7	
				3	3	1	11	4	
						2	7	9	
						3	12	1	
				4	2	1	6	9	{(6, 6, 2), (6, 7, 3)}
						2	8	2	
				5	2	1	13	10	
						2	15	5	
				6	2	1	3	4	{(6, 8, 3), (6, 9, 4)}
						2	13	3	
				7	3	1	10	15	
						2	8	6	
						3	11	1	
				8	3	1	10	7	
						2	9	2	
						3	14	13	
				9	4	1	13	2	
						2	7	12	
						3	12	6	
						4	13	5	
7	C	89	26	1	2	1	4	14	{(7, 2, 1), (7, 3, 2), (7, 4, 1)}
						2	13	3	
				2	1	1	5	7	{(7, 5, 1), (7, 6, 2)}
				3	2	1	6	10	{(7, 7, 1), (7, 8, 1)}
						2	8	3	
				4	1	1	2	9	{(7, 9, 1), (7, 10, 1), (7, 11, 1)}
				5	1	1	4	7	{(7, 12, 4), (7, 13, 2)}
				6	2	1	2	8	{(7, 14, 2), (7, 15, 3), (7, 16, 3)}
						2	7	5	
				7	1	1	3	3	{(7, 17, 2), (7, 18, 2)}
				8	1	1	5	6	{(7, 19, 4), (7, 20, 2)}
				9	1	1	6	5	{(7, 21, 5), (7, 22, 2)}
				10	1	1	1	7	{(7, 23, 2), (7, 24, 3)}
				11	1	1	3	3	{(7, 25, 4), (7, 26, 2)}
				12	4	1	9	8	
						2	13	1	
						3	12	4	

Table A.24 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	10	3	
				13	2	1	7	12	
						2	14	2	
				14	2	1	8	10	
						2	9	5	
				15	3	1	15	7	
						2	7	1	
						3	12	8	
				16	3	1	11	5	
						2	14	3	
						3	13	9	
				17	2	1	13	4	
						2	10	13	
				18	2	1	12	9	
						2	15	2	
				19	4	1	10	1	
						2	13	14	
						3	8	2	
						4	7	6	
				20	2	1	15	13	
						2	14	1	
				21	5	1	14	2	
						2	11	3	
						3	10	5	
						4	15	5	
						5	8	6	
				22	2	1	9	15	
						2	7	4	
				23	2	1	12	11	
						2	11	3	
				24	3	1	13	7	
						2	10	2	
						3	9	8	
				25	4	1	7	1	
						2	11	10	
						3	13	2	
						4	14	9	
				26	2	1	15	8	
						2	10	1	

Table A.25 Data of problem P25

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	36	8	1	1	1	1	10	{(1, 2, 5), (1, 3, 2), (1, 4, 3), (1, 5, 3), (1, 6, 4), (1, 7, 2), (1, 8, 4)}
				2	5	1	11	1	
						2	13	2	
						3	10	11	
						4	15	3	

Table A.25 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						5	12	8	
				3	2	1	13	15	
						2	9	1	
				4	3	1	12	3	
						2	7	7	
						3	11	9	
				5	3	1	15	11	
						2	10	2	
						3	8	6	
				6	4	1	14	10	
						2	7	3	
						3	8	2	
						4	15	3	
				7	2	1	9	14	
						2	13	3	
				8	4	1	7	6	
						2	8	3	
						3	10	8	
						4	14	9	
2	F	65	12	1	1	1	5	15	{(2, 2, 5), (2, 3, 4), (2, 4, 3), (2, 5, 4), (2, 6, 2), (2, 7, 2), (2, 8, 3), (2, 9, 4), (2, 10, 5), (2, 11, 3), (2, 12, 2)}
				2	5	1	8	7	
						2	12	2	
						3	13	14	
						4	9	3	
						5	14	2	
				3	4	1	10	7	
						2	13	4	
						3	7	1	
						4	8	13	
				4	3	1	14	2	
						2	10	9	
						3	13	1	
				5	4	1	9	3	
						2	8	6	
						3	14	4	
						4	7	10	
				6	2	1	11	13	
						2	15	3	
				7	2	1	10	12	
						2	12	4	
				8	3	1	7	7	
						2	9	8	
						3	15	1	
				9	4	1	15	13	
						2	10	3	
						3	12	8	
						4	11	4	

Table A.25 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				10	5	1	12	12	
						2	13	4	
						3	9	2	
						4	7	9	
						5	14	1	
				11	3	1	15	6	
						2	8	10	
						3	10	7	
				12	2	1	13	11	
						2	9	9	
3	T	41	5	1	2	1	5	9	{(3, 2, 1), (3, 3, 4)}
						2	9	4	
				2	1	1	6	6	{(3, 4, 3), (3, 5, 2)}
				3	4	1	13	8	
						2	8	4	
						3	14	13	
						4	10	3	
				4	3	1	8	5	
						2	7	14	
						3	15	1	
				5	2	1	11	9	
						2	9	3	
4	T	92	11	1	1	1	1	10	{(4, 2, 1), (4, 3, 3)}
				2	1	1	4	5	{(4, 4, 1), (4, 5, 3)}
				3	3	1	8	1	
						2	7	2	
						3	15	13	
				4	1	1	2	8	{(4, 6, 1), (4, 7, 5)}
				5	3	1	11	5	
						2	9	14	
						3	10	3	
				6	1	1	5	7	{(4, 8, 1), (4, 9, 2)}
				7	5	1	10	11	
						2	13	8	
						3	14	9	
						4	9	4	
						5	12	6	
				8	1	1	3	6	{(4, 10, 3), (4, 11, 4)}
				9	2	1	7	15	
						2	11	9	
				10	3	1	13	5	
						2	8	11	
						3	9	1	
				11	4	1	15	3	
						2	7	5	
						3	12	4	
						4	14	7	
5	C	66	12	1	1	1	1	9	{(5, 2, 1), (5, 3, 1), (5, 4, 1)}
				2	1	1	2	7	{(5, 5, 3), (5, 6, 2), (5, 7, 4)}
				3	1	1	3	5	{(5, 8, 3), (5, 9, 5)}

Table A.25 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				4	1	1	4	10	{(5, 10, 2), (5, 11, 3), (5, 12, 2)}
				5	3	1	9	11	
						2	10	2	
						3	13	4	
				6	2	1	12	5	
						2	7	8	
				7	4	1	15	3	
						2	11	13	
						3	14	1	
						4	8	5	
				8	3	1	14	6	
						2	15	7	
						3	10	12	
				9	5	1	7	8	
						2	10	7	
						3	9	2	
						4	11	3	
						5	13	10	
				10	2	1	8	15	
						2	12	3	
				11	3	1	7	4	
						2	15	14	
						3	9	3	
				12	2	1	10	6	
						2	14	9	
6	C	100	15	1	1	1	2	13	{(6, 2, 1), (6, 3, 1), (6, 4, 1), (6, 5, 1)}
				2	1	1	3	6	{(6, 6, 5), (6, 7, 3)}
				3	1	1	1	4	{(6, 8, 2), (6, 9, 3)}
				4	1	1	4	8	{(6, 10, 5), (6, 11, 2), (6, 12, 3)}
				5	1	1	5	9	{(6, 13, 4), (6, 14, 2), (6, 15, 2)}
				6	5	1	7	1	
						2	15	3	
						3	11	8	
						4	10	15	
						5	12	2	
				7	3	1	9	4	
						2	8	12	
						3	13	5	
				8	2	1	14	4	
						2	7	13	
				9	3	1	10	7	
						2	12	5	
						3	9	11	
				10	5	1	15	11	
						2	8	4	
						3	7	5	
						4	10	8	
						5	11	1	
				11	2	1	14	2	

Table A.25 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	12	10	
				12	3	1	8	4	
						2	9	5	
						3	13	4	
				13	4	1	12	4	
						2	7	6	
						3	10	4	
						4	15	3	
				14	2	1	13	10	
						2	14	9	
				15	2	1	11	8	
						2	9	3	
7	C	114	27	1	1	1	5	12	{(7, 2, 1), (7, 3, 1), (7, 4, 1)}
				2	1	1	1	6	{(7, 5, 1), (7, 6, 1)}
				3	1	1	6	8	{(7, 7, 1), (7, 8, 1)}
				4	1	1	4	7	{(7, 9, 1), (7, 10, 1), (7, 11, 1)}
				5	1	1	3	5	{(7, 12, 2), (7, 13, 3)}
				6	1	1	5	8	{(7, 14, 3), (7, 15, 4), (7, 16, 2)}
				7	1	1	4	4	{(7, 17, 2), (7, 18, 3)}
				8	1	1	2	6	{(7, 19, 3), (7, 20, 4)}
				9	1	1	6	3	{(7, 21, 2), (7, 22, 2)}
				10	1	1	2	5	{(7, 23, 2), (7, 24, 3)}
				11	1	1	1	7	{(7, 25, 2), (7, 26, 3), (7, 27, 2)}
				12	2	1	12	10	
						2	7	6	
				13	3	1	9	8	
						2	10	3	
						3	13	12	
				14	3	1	11	3	
						2	12	4	
						3	9	8	
				15	4	1	7	5	
						2	15	11	
						3	13	6	
						4	14	9	
				16	2	1	8	7	
						2	10	2	
				17	2	1	9	14	
						2	8	4	
				18	3	1	10	10	
						2	14	3	
						3	15	5	
				19	3	1	14	2	
						2	7	8	
						3	11	9	
				20	4	1	13	6	
						2	15	4	
						3	8	15	
						4	12	2	
				21	2	1	9	8	

Table A.25 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	11	12	
				22	2	1	13	3	
						2	14	5	
				23	2	1	8	3	
						2	7	6	
				24	3	1	15	5	
						2	12	7	
						3	10	1	
				25	2	1	7	13	
						2	8	7	
				26	3	1	10	3	
						2	13	9	
						3	9	6	
				27	2	1	12	2	
						2	15	9	

Table A.26 Data of problem P26

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	88	8	1	2	1	2	7	{(1, 2, 2), (1, 3, 4), (1, 4, 3), (1, 5, 5), (1, 6, 5), (1, 7, 3), (1, 8, 3)}
						2	7	5	
				2	2	1	8	2	
						2	10	6	
				3	4	1	10	1	
						2	9	7	
						3	15	9	
						4	11	3	
				4	3	1	9	2	
						2	7	3	
						3	14	12	
				5	5	1	11	2	
						2	14	8	
						3	7	6	
						4	13	7	
						5	8	9	
				6	5	1	13	3	
						2	15	12	
						3	9	6	
						4	10	3	
						5	7	1	
				7	3	1	9	15	
						2	8	2	
						3	12	1	
				8	3	1	13	4	
						2	10	7	
						3	9	8	

Table A.26 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
2	F	92	12	1	2	1	6	14	{(2, 2, 2), (2, 3, 4), (2, 4, 5), (2, 5, 3), (2, 6, 5), (2, 7, 4), (2, 8, 2), (2, 9, 2), (2, 10, 3), (2, 11, 5), (2, 12, 2)}
						2	13	8	
				2	2	1	9	11	
						2	12	7	
				3	4	1	12	5	
						2	8	4	
						3	13	12	
						4	15	1	
				4	5	1	12	6	
						2	7	10	
						3	14	1	
						4	15	3	
						5	8	7	
				5	3	1	9	6	
						2	10	15	
						3	7	5	
				6	5	1	13	9	
						2	8	7	
						3	9	14	
						4	10	4	
						5	14	5	
				7	4	1	10	8	
						2	15	2	
						3	9	10	
						4	13	3	
				8	2	1	14	13	
						2	13	2	
				9	2	1	7	3	
						2	10	9	
				10	3	1	8	3	
						2	10	11	
						3	11	7	
				11	5	1	15	6	
						2	11	2	
						3	7	10	
						4	13	1	
						5	9	2	
				12	2	1	8	9	
						2	15	4	
3	T	68	5	1	2	1	1	10	{(3, 2, 2), (3, 3, 4)}
						2	7	5	
				2	2	1	2	4	{(3, 4, 4), (3, 5, 3)}
						2	15	3	
				3	4	1	10	2	
						2	9	15	
						3	11	2	
						4	8	1	

Table A.26 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				4	4	1	8	9	
						2	7	2	
						3	10	4	
						4	11	8	
				5	3	1	13	7	
						2	15	8	
						3	7	3	
4	T	81	11	1	1	1	6	9	{(4, 2, 2), (4, 3, 4)}
				2	2	1	5	11	{(4, 4, 1), (4, 5, 2)}
						2	7	8	
				3	4	1	8	6	
						2	14	12	
						3	12	5	
						4	10	9	
				4	1	1	4	10	{(4, 6, 2), (4, 7, 4)}
				5	2	1	11	9	
						2	7	4	
				6	2	1	3	6	{(4, 8, 1), (4, 9, 5)}
						2	13	11	
				7	4	1	13	2	
						2	8	7	
						3	7	3	
						4	9	10	
				8	1	1	2	5	{(4, 10, 3), (4, 11, 3)}
				9	5	1	7	4	
						2	14	5	
						3	13	3	
						4	15	11	
						5	10	3	
				10	3	1	9	7	
						2	8	5	
						3	12	4	
				11	3	1	10	9	
						2	11	1	
						3	15	2	
5	C	48	12	1	1	1	5	12	{(5, 2, 1), (5, 3, 1), (5, 4, 1)}
				2	1	1	6	8	{(5, 5, 3), (5, 6, 4)}
				3	1	1	1	6	{(5, 7, 3), (5, 8, 3)}
				4	1	1	2	7	{(5, 9, 2), (5, 10, 2), (5, 11, 4), (5, 12, 5)}
				5	3	1	8	3	
						2	11	8	
						3	13	7	
				6	4	1	9	1	
						2	10	6	
						3	7	10	
						4	14	9	
				7	3	1	12	2	
						2	9	13	
						3	7	4	

Table A.26 (Continued)

p	*PT*	D_p	n_p	*i*	J_{pi}	*j*	*m*	t_{pijm}	E_{pij}
				8	3	1	15	3	
						2	8	11	
						3	10	5	
				9	2	1	13	10	
						2	14	1	
				10	2	1	7	4	
						2	15	12	
				11	4	1	11	3	
						2	12	14	
						3	13	1	
						4	9	4	
				12	5	1	14	5	
						2	8	4	
						3	9	2	
						4	10	15	
						5	7	3	
6	C	96	16	1	1	1	2	13	{(6, 2, 2), (6, 3, 2), (6, 4, 2), (6, 5, 2)}
				2	2	1	3	10	{(6, 6, 2), (6, 7, 3), (6, 8, 2)}
						2	15	4	
				3	2	1	4	5	{(6, 9, 3), (6, 10, 5)}
						2	9	5	
				4	2	1	5	4	{(6, 11, 4), (6, 12, 3)}
						2	7	6	
				5	2	1	6	11	{(6, 13, 2), (6, 14, 3), (6, 15, 2), (6, 16, 3)}
						2	13	4	
				6	2	1	10	9	
						2	12	10	
				7	3	1	15	11	
						2	13	9	
						3	9	1	
				8	2	1	8	2	
						2	7	8	
				9	3	1	9	14	
						2	15	3	
						3	13	6	
				10	5	1	7	5	
						2	8	7	
						3	12	4	
						4	14	8	
						5	11	1	
				11	4	1	13	1	
						2	12	3	
						3	14	15	
						4	10	4	
				12	3	1	8	7	
						2	9	8	
						3	11	4	
				13	2	1	13	5	

Table A.26 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	10	7	
				14	3	1	12	5	
						2	7	13	
						3	14	2	
				15	2	1	14	1	
						2	8	12	
				16	3	1	11	10	
						2	9	4	
						3	15	6	
7	C	89	20	1	2	1	6	11	{(7, 2, 1), (7, 3, 2)}
						2	13	4	
				2	1	1	1	10	{(7, 4, 1), (7, 5, 1), (7, 6, 1)}
				3	2	1	2	8	{(7, 7, 1), (7, 8, 1)}
						2	7	5	
				4	1	1	3	3	{(7, 9, 3), (7, 10, 4)}
				5	1	1	4	8	{(7, 11, 2), (7, 12, 5)}
				6	1	1	5	5	{(7, 13, 3), (7, 14, 2)}
				7	1	1	6	9	{(7, 15, 2), (7,16, 2), (7, 17, 3)}
				8	1	1	1	4	{(7, 18, 2), (7, 19, 4), (7, 20, 3)}
				9	3	1	7	3	
						2	10	6	
						3	8	4	
				10	4	1	15	8	
						2	14	2	
						3	11	2	
						4	13	10	
				11	2	1	10	2	
						2	7	12	
				12	5	1	8	1	
						2	12	10	
						3	14	3	
						4	11	8	
						5	15	4	
				13	3	1	9	15	
						2	13	2	
						3	12	1	
				14	2	1	11	9	
						2	14	8	
				15	2	1	13	15	
						2	9	3	
				16	2	1	14	2	
						2	8	14	
				17	3	1	7	3	
						2	10	13	
						3	11	5	
				18	2	1	9	4	
						2	15	10	
				19	4	1	10	7	
						2	7	2	
						3	14	11	

Table A.26 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	12	6	
				20	3	1	8	5	
						2	11	14	
						3	13	1	

Table A.27 Data of problem P27

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	64	5	1	1	1	3	8	{(1, 2, 3), (1, 3, 4), (1, 4, 3), (1, 5, 5)}
				2	3	1	11	2	
						2	9	13	
						3	10	6	
				3	4	1	13	4	
						2	9	1	
						3	15	3	
						4	12	9	
				4	3	1	7	5	
						2	13	3	
						3	8	11	
				5	5	1	12	2	
						2	10	8	
						3	14	10	
						4	7	1	
						5	11	3	
2	F	53	6	1	1	1	5	12	{(2, 2, 5), (2, 3, 3), (2, 4, 2), (2, 5, 3), (2, 6, 4)}
				2	5	1	8	4	
						2	13	2	
						3	11	8	
						4	14	3	
						5	10	3	
				3	3	1	11	5	
						2	13	13	
						3	12	1	
				4	2	1	7	15	
						2	15	2	
				5	3	1	10	8	
						2	13	11	
						3	12	2	
				6	4	1	12	6	
						2	15	9	
						3	14	5	
						4	9	3	
3	F	35	9	1	1	1	5	9	{(3, 2, 5), (3, 3, 4), (3, 4, 2), (3, 5, 3), (3, 6, 4), (3, 7, 5), (3, 8, 2), (3, 9, 3)}
				2	5	1	10	3	
						2	7	2	

Table A.27 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	11	6	
						4	13	8	
						5	14	3	
				3	4	1	14	1	
						2	8	4	
						3	9	2	
						4	10	10	
				4	2	1	9	11	
						2	7	9	
				5	3	1	13	2	
						2	8	7	
						3	12	5	
				6	4	1	7	2	
						2	9	4	
						3	15	13	
						4	8	1	
				7	5	1	8	2	
						2	14	7	
						3	11	3	
						4	7	6	
						5	10	8	
				8	2	1	15	1	
						2	9	15	
				9	3	1	7	9	
						2	12	7	
						3	11	5	
4	F	48	11	1	1	1	3	12	{(4, 2, 3), (4, 3, 4), (4, 4, 5), (4, 5, 5), (4, 6, 3), (4, 7, 2), (4, 8, 2), (4, 9, 4), (4, 10, 3), (4, 11, 2)}
				2	3	1	9	15	
						2	7	4	
						3	14	1	
				3	4	1	8	5	
						2	11	3	
						3	10	14	
						4	12	2	
				4	5	1	12	6	
						2	9	3	
						3	14	9	
						4	13	11	
						5	15	1	
				5	5	1	11	4	
						2	15	9	
						3	10	7	
						4	7	13	
						5	13	3	
				6	3	1	14	7	
						2	9	2	
						3	8	9	

Table A.27 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				7	2	1	15	12	
						2	10	6	
				8	2	1	13	2	
						2	11	10	
				9	4	1	10	7	
						2	11	13	
						3	8	8	
						4	14	3	
				10	3	1	15	2	
						2	13	10	
						3	7	9	
				11	2	1	7	9	
						2	9	8	
5	T	60	5	1	1	1	1	5	{(5, 2, 1), (5, 3, 4)}
				2	1	1	2	7	{(5, 4, 5), (5, 5, 3)}
				3	4	1	7	11	
						2	11	3	
						3	14	5	
						4	9	2	
				4	5	1	12	9	
						2	7	4	
						3	9	6	
						4	10	7	
						5	13	2	
				5	3	1	8	12	
						2	11	3	
						3	15	1	
6	T	40	9	1	1	1	1	9	{(6, 2, 1), (6, 3, 4)}
				2	1	1	2	4	{(6, 4, 1), (6, 5, 2)}
				3	4	1	11	15	
						2	7	2	
						3	8	3	
						4	10	7	
				4	1	1	3	5	{(6, 6, 1), (6, 7, 3)}
				5	2	1	12	9	
						2	9	6	
				6	1	1	4	7	{(6, 8, 2), (6, 9, 3)}
				7	3	1	14	4	
						2	8	14	
						3	12	1	
				8	2	1	7	13	
						2	13	2	
				9	3	1	10	3	
						2	11	1	
						3	15	11	
7	T	164	11	1	2	1	3	13	{(7, 2, 1), (7, 3, 3)}
						2	9	5	
				2	1	1	5	4	{(7, 4, 2), (7, 5, 5)}
				3	3	1	14	11	
						2	7	3	

Table A.27 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	13	9	
				4	2	1	2	9	{(7, 6, 1), (7, 7, 3)}
						2	12	4	
				5	5	1	15	7	
						2	10	5	
						3	9	2	
						4	11	6	
						5	8	10	
				6	1	1	4	12	{(7, 8, 2), (7, 9, 2)}
				7	3	1	7	1	
						2	9	13	
						3	14	5	
				8	2	1	1	8	{(7, 10, 4), (7, 11, 2)}
						2	7	3	
				9	2	1	11	5	
						2	13	4	
				10	4	1	8	1	
						2	15	2	
						3	14	6	
						4	10	15	
				11	2	1	9	7	
						2	15	14	
8	T	142	13	1	1	1	2	10	{(8, 2, 1), (8, 3, 3)}
				2	1	1	1	7	{(8, 4, 1), (8, 5, 2)}
				3	3	1	13	10	
						2	10	5	
						3	15	1	
				4	1	1	4	9	{(8, 6, 1), (8, 7,4)}
				5	2	1	8	7	
						2	12	14	
				6	1	1	6	5	{(8, 8, 1), (8, 9, 5)}
				7	4	1	15	12	
						2	13	7	
						3	11	3	
						4	9	1	
				8	1	1	5	11	{(8, 10, 1), (8, 11, 3)}
				9	5	1	9	6	
						2	11	3	
						3	7	9	
						4	10	2	
						5	8	4	
				10	1	1	3	6	{(8, 12, 2), (8, 13, 4)}
				11	3	1	10	11	
						2	15	4	
						3	13	6	
				12	2	1	8	15	
						2	9	8	
				13	4	1	14	4	
						2	11	13	
						3	13	1	

Table A.27 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						4	7	3	
9	C	52	12	1	1	1	3	8	{(9, 2, 2), (9, 3, 2), (9, 4, 2)}
				2	2	1	5	7	{(9, 5, 5), (9, 6, 2)}
						2	9	3	
				3	2	1	6	10	{(9, 7, 3), (9, 8, 4), (9, 9, 2)}
						2	12	2	
				4	2	1	2	6	{(9, 10, 2), (9, 11, 5), (9, 12, 2)}
						2	7	5	
				5	5	1	11	6	
						2	9	4	
						3	15	11	
						4	7	6	
						5	14	7	
				6	2	1	12	10	
						2	10	3	
				7	3	1	7	8	
						2	14	3	
						3	13	12	
				8	4	1	8	10	
						2	12	1	
						3	9	13	
						4	11	2	
				9	2	1	15	5	
						2	10	1	
				10	2	1	14	11	
						2	12	5	
				11	5	1	10	15	
						2	13	3	
						3	15	8	
						4	11	2	
						5	8	4	
				12	2	1	9	8	
						2	7	14	
10	C	90	22	1	1	1	1	10	{(10, 2, 1), (10, 3, 1), (10, 4, 1)}
				2	1	1	6	7	{(10, 5, 1), (10, 6, 1)}
				3	1	1	2	9	{(10, 7, 1), (10, 8, 1)}
				4	1	1	5	11	{(10, 9, 1), (10, 10, 1)}
				5	1	1	4	5	{(10, 11, 2), (10, 12, 3)}
				6	1	1	3	4	{(10, 13, 3), (10, 14, 4)}
				7	1	1	1	3	{(10, 15, 2), (10, 16, 2)}
				8	1	1	5	8	{(10, 17, 5), (10, 18, 2)}
				9	1	1	2	7	{(10, 19, 3), (10, 20, 4)}
				10	1	1	6	5	{(10, 21, 3), (10, 22, 2)}
				11	2	1	7	3	
						2	12	8	
				12	3	1	8	2	
						2	10	8	
						3	11	7	
				13	3	1	15	13	
						2	14	5	

Table A.27 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						3	9	6	
				14	4	1	10	4	
						2	11	2	
						3	8	3	
						4	13	10	
				15	2	1	11	12	
						2	10	2	
				16	2	1	13	3	
						2	14	8	
				17	5	1	14	1	
						2	9	11	
						3	12	7	
						4	13	6	
						5	7	8	
				18	2	1	15	9	
						2	8	3	
				19	3	1	9	1	
						2	11	2	
						3	14	10	
				20	4	1	12	3	
						2	7	6	
						3	8	15	
						4	15	1	
				21	3	1	13	7	
						2	11	3	
						3	10	2	
				22	2	1	7	14	
						2	9	6	

Table A.28 Data of problem P28

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
1	F	45	5	1	2	1	4	7	{(1, 2, 4), (1, 3, 2), (1, 4, 4), (1, 5, 3)}
						2	15	3	
				2	4	1	11	14	
						2	8	7	
						3	7	6	
						4	13	8	
				3	2	1	13	15	
						2	10	4	
				4	4	1	7	2	
						2	15	5	
						3	12	13	
						4	8	4	
				5	3	1	14	9	
						2	9	8	
						3	11	2	

Table A.28 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
2	F	51	6	1	2	1	6	7	{(2, 2, 2), (2, 3, 4), (2, 4, 3), (2, 5, 5), (2, 6, 2)}
						2	9	4	
				2	2	1	11	15	
						2	8	2	
				3	4	1	13	2	
						2	9	9	
						3	7	6	
						4	14	3	
				4	3	1	9	1	
						2	13	10	
						3	10	5	
				5	5	1	15	4	
						2	7	3	
						3	11	7	
						4	9	8	
						5	12	1	
				6	2	1	8	11	
						2	7	3	
3	F	44	8	1	2	1	2	7	{(3, 2, 2), (3, 3, 4), (3, 4, 3), (3, 5, 5), (3, 6, 5), (3, 7, 3), (3, 8, 3)}
						2	7	5	
				2	2	1	8	2	
						2	10	6	
				3	4	1	10	1	
						2	9	7	
						3	15	9	
						4	11	3	
				4	3	1	9	2	
						2	7	3	
						3	14	12	
				5	5	1	11	2	
						2	14	8	
						3	7	6	
						4	13	7	
						5	8	9	
				6	5	1	13	3	
						2	15	12	
						3	9	6	
						4	10	3	
						5	7	1	
				7	3	1	9	15	
						2	8	2	
						3	12	1	
				8	3	1	13	4	
						2	10	7	
						3	9	8	

Table A.28 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
4	F	98	11	1	2	1	4	14	{(4, 2, 2), (4, 3, 4), (4, 4, 3), (4, 5, 3), (4, 6, 4), (4, 7, 2), (4, 8, 5), (4, 9, 4), (4, 10, 2), (4, 11, 2)}
						2	11	2	
				2	2	1	8	11	
						2	10	9	
				3	4	1	13	6	
						2	9	5	
						3	10	10	
						4	14	2	
				4	3	1	14	8	
						2	8	4	
						3	13	12	
				5	3	1	7	3	
						2	9	15	
						3	15	9	
				6	4	1	12	4	
						2	11	15	
						3	7	2	
						4	8	1	
				7	2	1	7	14	
						2	11	1	
				8	5	1	7	6	
						2	14	9	
						3	8	1	
						4	15	14	
						5	9	3	
				9	4	1	15	8	
						2	11	10	
						3	10	1	
						4	7	13	
				10	2	1	9	3	
						2	12	8	
				11	2	1	10	9	
						2	8	2	
5	T	56	5	1	1	1	3	8	{(5, 2, 2), (5, 3, 5)}
				2	2	1	4	8	{(5, 4, 2), (5, 5, 3)}
						2	8	3	
				3	5	1	15	2	
						2	14	9	
						3	7	4	
						4	12	6	
						5	10	1	
				4	2	1	9	12	
						2	13	5	
				5	3	1	11	8	
						2	8	7	
						3	14	3	
6	T	64	9	1	1	1	5	11	{(6, 2, 2), (6, 3, 5)}

Table A.28 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				2	2	1	6	8	{(6, 4, 1), (6, 5, 3)}
						2	14	3	
				3	5	1	15	1	
						2	7	9	
						3	13	5	
						4	9	4	
						5	10	8	
				4	1	1	1	6	{(6, 6, 2), (6, 7, 2)}
				5	3	1	14	7	
						2	10	11	
						3	11	2	
				6	2	1	2	7	{(6, 8, 4), (6, 9, 2)}
						2	9	6	
				7	2	1	11	10	
						2	8	5	
				8	4	1	9	5	
						2	10	13	
						3	11	2	
						4	7	3	
				9	2	1	14	8	
						2	8	5	
7	T	114	11	1	1	1	3	7	{(7, 2, 2), (7, 3, 5)}
				2	2	1	6	5	{(7, 4, 2), (7, 5, 3)}
						2	7	3	
				3	5	1	10	5	
						2	13	15	
						3	15	5	
						4	11	4	
						5	14	11	
				4	2	1	2	10	{(7, 6, 2), (7, 7, 4)}
						2	12	6	
				5	3	1	9	4	
						2	13	2	
						3	8	12	
				6	2	1	4	9	{(7, 8, 2), (7, 9, 5)}
						2	15	7	
				7	4	1	15	10	
						2	9	1	
						3	11	3	
						4	7	3	
				8	2	1	1	6	{(7, 10, 2), (7, 11, 2)}
						2	9	3	
				9	5	1	11	14	
						2	14	9	
						3	10	1	
						4	13	2	
						5	8	3	
				10	2	1	7	1	
						2	10	13	
				11	2	1	13	8	

Table A.28 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	15	2	
8	T	180	13	1	1	1	6	7	{(8, 2, 2), (8, 3, 5)}
				2	2	1	5	6	{(8, 4, 1), (8, 5, 3)}
						2	12	6	
				3	5	1	13	10	
						2	10	13	
						3	14	3	
						4	8	2	
						5	9	5	
				4	1	1	4	13	{(8, 6, 2), (8, 7, 4)}
				5	3	1	12	3	
						2	15	6	
						3	7	15	
				6	2	1	3	10	{(8, 8, 1), (8, 9, 2)}
						2	8	3	
				7	4	1	9	7	
						2	7	3	
						3	13	1	
						4	11	9	
				8	1	1	2	8	{(8, 10, 2), (8, 11, 2)}
				9	2	1	14	12	
						2	10	9	
				10	2	1	1	5	{(8, 12, 3), (8, 13, 4)}
						2	9	14	
				11	2	1	8	2	
						2	11	7	
				12	3	1	15	11	
						2	9	4	
						3	10	2	
				13	4	1	11	4	
						2	14	1	
						3	12	8	
						4	7	5	
9	C	48	13	1	2	1	1	12	{(9, 2, 1), (9, 3, 2), (9, 4, 1)}
						2	13	3	
				2	1	1	4	9	{(9, 5, 3), (9, 6, 4), (9, 7, 2), (9, 8, 3)}
				3	2	1	5	3	{(9, 9, 2), (9, 10, 5)}
						2	7	4	
				4	1	1	2	7	{(9, 11, 3), (9, 12, 2), (9, 13, 4)}
				5	3	1	7	13	
						2	14	4	
						3	11	1	
				6	4	1	8	4	
						2	12	3	
						3	9	2	
						4	10	12	
				7	2	1	14	9	
						2	9	4	
				8	3	1	13	11	

Table A.28 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
						2	8	2	
						3	15	6	
				9	2	1	9	15	
						2	8	5	
				10	5	1	15	1	
						2	10	7	
						3	14	3	
						4	12	10	
						5	7	5	
				11	3	1	9	3	
						2	8	14	
						3	13	4	
				12	2	1	11	10	
						2	15	8	
				13	4	1	10	5	
						2	7	2	
						3	14	11	
						4	12	6	
10	C	53	17	1	2	1	4	10	{(10, 2, 1), (10, 3, 2), (10, 4, 2), (10, 5, 2)}
						2	8	5	
				2	1	1	5	8	{(10, 6, 2), (10, 7, 3), (10, 8, 3)}
				3	2	1	6	6	{(10, 9, 3), (10, 10, 4), (10, 11, 2)}
						2	7	3	
				4	2	1	1	11	{(10, 12, 2), (10, 13, 2), (10, 14, 3), (10, 15, 4)}
						2	9	2	
				5	2	1	2	3	{(10, 16, 5), (10, 17, 2)}
						2	14	4	
				6	2	1	15	10	
						2	12	2	
				7	3	1	10	3	
						2	7	15	
						3	9	1	
				8	3	1	11	8	
						2	8	2	
						3	14	5	
				9	3	1	7	1	
						2	15	3	
						3	10	12	
				10	4	1	13	12	
						2	14	4	
						3	12	5	
						4	11	1	
				11	2	1	9	14	
						2	8	6	
				12	2	1	8	14	
						2	7	4	
				13	2	1	12	4	
						2	15	13	

Table A.28 (Continued)

p	PT	D_p	n_p	i	J_{pi}	j	m	t_{pijm}	E_{pij}
				14	3	1	14	10	
						2	10	8	
						3	13	3	
				15	4	1	15	6	
						2	9	4	
						3	12	11	
						4	11	1	
				16	5	1	7	2	
						2	15	7	
						3	8	3	
						4	11	10	
						5	12	9	
				17	2	1	10	6	
						2	9	8	

LIST OF PUBLICATIONS

1. **Dileeplal, J.**, Narayanan, K. P., Sudheer, C. B., and Girish, B. S. (2012) Pareto archived genetic algorithm for multi-objective assembly job shop scheduling problem, *Proceedings of the International Conference on Mathematical Modeling and Applied Soft Computing,* CIT Coimbatore, 1177-1184.

2. **Dileeplal, J.**, Sudheer, C. B. and Girish, B. S. (2011) Multi-objective tabu search for assembly job shop scheduling problem, *Proceedings of the International Conference on Mathematical Modelling and Applications to Industrial Problems,* NIT Calicut, 296-302.

3. **Dileeplal, J.**, Girish, B. S. and Sudheer, C. B. (2010) Multi-objective genetic algorithm for assembly job shop scheduling problem, *Proceedings of the XIV Annual International Conference of the Society of Operations Management*, NITIE Mumbai (in CD).

4. **Dileeplal, J.**, Girish, B. S., Narayanan, K. P. and Sudheer, C. B. Multi-objective assembly job shop scheduling based on Pareto approach using genetic algorithm and tabu search. (Under preparation)

CURRICULUM VITAE

DILEEPLAL J.

Official Address:
Associate Professor,
Department of Mechanical Engineering,
Mar Athanasius College of Engineering,
Kothamangalam,
Kerala, India, Pin – 686 666

Residential Address:
Agape,
Perumaram,
Vizhinjam P. O.,
Thiruvananthapuram,
Kerala, India, Pin – 695 521

E-mail: dileeplal@gmail.com

Mob: 09447587714

Education

Ph. D.
Department of Ship Technology
Cochin University of Science and Technology.

M. Tech. in Engineering Statistics (First Class with Distinction and Second Rank), 2004
Department of Statistics
Cochin University of Science and Technology.

B. Tech. in Production Engineering (First Class), 2001
Govt. Engineering College Thrissur
University of Calicut.

Professional Experience

Designation:	Associate Professor
Duration:	June 14, 2012 – still continuing
Designation:	Assistant Professor
Duration:	June 14, 2009 – June 13, 2012
Designation:	Lecturer
Duration:	June 14, 2004 – June 13, 2009
Organisation:	Mar Athanasius College of Engineering, Kothamangalam, Kerala, Pin - 686 666.

Membership in Professional Societies

- Indian Society for Technical Education - Life Member
- Society of Operations Management (India) - Life Member

Personal Information

Sex:	Male
Marital Status:	Married
Date of Birth:	31/05/1980
Nationality:	Indian
Mother Tongue:	Malayalam