Optimization of the Patients Appointments in Chemotherapy Treatment Unit: Heuristic and Metaheuristic Approaches

by

Sanjana Shahnawaz

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Mechanical and Manufacturing Engineering

University of Manitoba

Winnipeg

Abstract

This research aims to improve the performance of the service of a Chemotherapy Treatment Unit by reducing the waiting time of patients within the unit. In order to fulfill the objective, initially, the chemotherapy treatment unit is deduced as an identical parallel machines scheduling problem with unequal release time and single resource. A mathematical model is developed to generate the optimum schedule. Afterwards, a Tabu search (TS) algorithm is developed. The performance of the TS algorithm is evaluated by comparing results with the mathematical model and the best results of benchmark problems reported in the literature. Later on, an additional resource is considered which converted the problem into a dual resources scheduling problem. Three approaches are proposed to solve this problem; namely, heuristics, a Tabu search algorithm with heuristic (TSHu), and Tabu search algorithm for dual resources (TSD).

Acknowledgments

First and foremost, I would like to express my gratitude to my thesis supervisors, Dr. Tarek ElMekkawy and Dr. Qingjin Peng for their support, guidance and supervision. Their help, advice and patience have been invaluable on both an academic and a personal level, for which I am extremely grateful.

I would like to thank Sue Bates, Director of Patient Navigation, CancerCare Manitoba for her continuous support. Finally, without the financial support of CancCare Manitoba, this research will not be achieved.

I also appreciate the co-operation of the committee members who attended the thesis defence.

Dedication

This thesis is dedicated to my mother for instilling the importance of hard work and higher education.

Contents

List of Tables

List of Figures

List of Copyrighted Material

Thesis of Ahmed Z., 2011, “Developing an Efficient Scheduling Template of a Chemotherapy Treatment Unit: Simulation and Optimization Approach”, University of Manitoba, Winnipeg, Manitoba, Canada.

List of Symbols

BA	Backward Algorithm
BL	Best List
CCMB	CancerCare Manitoba
CIHI	Canadian Institute for Health Information
CR	Critical Ratio
ERD	Earliest Release Date
EDD	Earliest Due Date
FIFO	First In First Out
GA	Genetic Algorithm
GDP	Gross Domestic Products
GSPT	Generalized Shortest Processing Time
Greedy SPT	Greedy Shortest Processing Time
ISArray	Initial Solutions Array
LMC	Largest Marginal Contribution
LPT	Largest Processing Time
LPT-LRD	Longest Processing Time & Longest Release Date
MFHA	Modified Forward Heuristic Algorithm
MIS	Multiple Initial Solutions
NP-hard	Non-deterministic Polynomial-time hard
PSO	Particle Swarm Optimization

SA	Simulated Annealing
SES	A heuristic based on the priority rules: SPT, EDD and Slack time
SPT	Shortest Processing Time
SPT-A_i	A heuristic based on Shortest Processing Time where A_i is the constant portion of processing that does not change with the resource allocation
SPT/L(east)PA	A heuristic based on Shortest Processing Time which takes the least processing time between the constant portion of processing time and the portion that changes with the resource allocation
SPT/L(argest)PA	A heuristic based on Shortest Processing Time which takes the largest processing time between the constant portion of processing time and the portion that changes with the resource allocation
SRPT	Shortest Remaining Processing Time
TL	Tabu List
TS	Tabu Search algorithm
TSD	Tabu Search algorithm for Dual resources
TSHu	Tabu Search Heuristic algorithm
TS+I	Tabu Search algorithm with Intensification

Chapter 1

Introduction

1.1 Background

Health care in Canada is delivered through a publically funded system. It is guided by the provisions of the Canada Health Act of 1984. The government assures the quality of care through federal standards. To maintain the excellence in the service, the government of Canada spends a significant amount of money in this sector. According to Canadian Institute for Health Information (CIHI), in 1975, total Canadian health care cost consumed 7% of the Gross Domestic Products (GDP). Canada's total expenditures as a percentage of GDP grew to 11.75% in 2010. Hospitals represent the largest category of public sector spending (37%) of the government. However, the report of CIHI (2010) implies, this proportion varies among provinces, from 30.4% in Quebec and 33.9% in Newfoundland and Labrador to 44.5% in Manitoba and 47.2 % in Nova Scotia. Moreover, total expenditure of Canada in 2011 has crossed \$200 billion which is \$7 billion more than the year of 2010. This amounts to roughly \$5800 per Canadian, about \$150 more per person than 2010. The spending in health care in 2011 has increased by 4.0% over a year.

The costs of health care are increasing day by day along with the population of the country. Manitoba finds itself among those provinces of Canada where the government spends large amount of money in the health care sector (Canadian Institute for Health Information, Health Canada). However, 40,000 Manitobans live with the Cancer (Manitoba Health Annual Report). It is predicted that by the end of 2025, more than 60,000 Manitobans will be living with this disease. Approximately 6000 peoples of Manitoba are diagnosed with cancer each year. To support this large number, an agency named CancerCare Manitoba, provincially legislated, is formed.

CancerCare Manitoba (CCMB), located in Winnipeg, Manitoba is responsible for providing care, treatment and support across the entire cancer service spectrum- from prevention, early detection, diagnosis, treatment and care, to palliation or end of life care. The agency operates from its main site, MacCharles. Among all the departments in the clinic Chemotherapy treatment unit is built with the aim of providing chemotherapy to cancer patients.

CCMB provides fully integrated service for hematology, radiation oncology, pediatric oncology, gynecologic oncology and genitourinary oncology. Patients are required to visit clinics for their chemotherapy treatment. The treatment duration varies depending on the nature of cancer and drugs and is ranges from an hour to 12 hours. CancerCare Manitoba, in order to accommodate the huge number of patients is undertaking variety of continuous improvement projects. Cancer is a burden experienced by patients and their care providers throughout the full trajectory of the disease. Challenges include a myriad of physical, social, emotional, nutritional, informational, psychological, spiritual, and practical needs (CancerCare Manitoba). In the current health care system, cancer patients report fragmentation of service, delays in access and inadequate information (CancerCare Manitoba). This study is carried out in order to improve the quality of service in the chemotherapy treatment unit through reducing the long waiting time of patients in the clinic.

Industrial engineering approaches are commonly used in the health care organizations for years. Generally care providers involve industrial engineering tools to ensure the quality of service employing core quality management techniques. However, researchers of engineering used to keep their study in the domain of manufacturing environment. But with the advancement of time, it is realized that similar approaches are also applicable in the health care system. The potential approaches include: scheduling, queuing and simulation, optimization, statistical quality control, management of information systems, process reengineering, continuous improvement and design and layout.

This study considers operational research problems including scheduling and optimization in the chemotherapy treatment unit of CancerCare Manitoba, MacCharles site. With the goal of minimizing the waiting time of patients by minimizing the flow time of the system, this study simplifies the dual resources scheduling problem into an identical parallel machines scheduling problem with single resource where release time of patients are not equal. This study proposes an efficient algorithm using a metaheuristic approach: Tabu search (TS) to schedule the identical parallel machines with single resource. Moreover, a mathematical model is developed to solve for the optimum solutions. Eventually, an additional resource is included as a constraint. Tabu search algorithm with heuristic (TSHu) and Tabu search algorithm for dual resources (TSD) are developed to deal with the dual resources' scheduling problem. Moreover, a simulation study is carried out using several dispatching rules.

To maintain and improve the excellence in service, an efficient scheduling template is required for the care providers to book appointments for patients. Accommodating a good number of patients within the time frame of the clinic hours with reduced waiting time of patients increases the complexity of scheduling. On the contrary, an efficient scheduling ensures proper resource utilization with proper service provided to the patients. It is an important issue to both the care providers and the care receivers.

1.2 Objective of the thesis

This study aims at developing an efficient scheduling template for the chemotherapy unit of CCMB by minimizing the flow time of patients. Minimization of flow time ensures minimum waiting time of patients in the clinic. The chemotherapy treatment unit consists of two scarce resources to provide the service: chairs and nurses. In the first part of the thesis, the problem is simplified as parallel machine problem with single resource (chair). A Tabu Search (TS) algorithm is proposed to obtain the optimal (or near optimal) schedule of this problem. In addition, a mathematical model is developed to obtain the optimal schedule of the same problem but for small problem size. The performance of the TS algorithm is measured by comparing its results with the ones reported in the literature and the optimal solutions of the mathematical model. The outcome of this study proves the superiority of the developed TS algorithm. Afterwards both resources (Nurses and Chairs) are considered for scheduling patients in the second part of the thesis. An extended experiment is conducted in this part with heuristic and metaheuristic (Tabu Search) approaches. Nevertheless, the arrival pattern of patients has an effect on the flow time. Therefore, a study is carried out with varying the release time of patients in both the parts of the thesis.

1.3 Thesis outline

Chapter 2 describes the related literature. In Chapter 3, the problem under consideration is treated as identical parallel machines scheduling problem with release time constraints. Chapter 4 introduces the proposed heuristics and metaheuristics that deal with dual resource problem. A case study of the real situation is presented in Chapter 4 as well. Chapter 5 concludes the thesis with the scope of future work.

Chapter 2

Literature Review

In order to develop an efficient scheduling template of the chemotherapy treatment unit, the proposed approaches of this thesis considered the problem as identical parallel machine. As it was mentioned in Chapter 1, the treatment unit has two scarce resources; namely nurses and chairs. The problem is solved considering the chairs only and later three approaches have been introduced to consider the availability of both resources simultaneously. Therefore, a detailed survey is carried on Identical Parallel Machines Scheduling problems that considered both the single resource and dual resources. Section 2.1 provides a review on identical parallel machines with just one resource. Section 2.2 summarizes the previous work done on identical parallel machines with dual resources.

2.1 Scheduling Identical Parallel Machines with Single Resource

In the healthcare unit where patients are served for chemotherapy, both the availability of a nurse and a chair is required for each patient. Whenever a nurse and a chair are free, the nurse brings a patient and infuses the line of drug into the patient's body. A patient occupies the chair throughout the treatment time. The time of treatment varies from patient to patient within a range of less than one hour to ten hours and the patients have different arrival times. A nurse again is needed to remove the drug line of the patient after the infusion. In the process the nurse can serve other patients who need the service of infusion or removal. Chairs can be used for any of the patients.

Thus, the environment of the chemotherapy unit is considered similar to the identical parallel machines scheduling problem where jobs have different release and processing times and are allowed to be processed on any of the machines (as the machines are identical). An extensive literature review is conducted here on identical parallel machines (with single resource: the chair) considering the release time constraint and with the objective of minimizing the total flow time of the system. According to the standard machine scheduling classification, this problem is denoted as, $Pm|r_i|\sum F_i$ where Pm indicates m number of Parallel machine, r_i is release time of job i and F_i is the flow time of job i.

Parallel machine scheduling has been a popular research area for researchers because of its wide range of potential application areas and this popularity has been increased considerably during recent years. The problem of parallel machine scheduling (PMS) with unequal release date is proved to be NP-hard by Lenstra and Kan (1977). The problem remains NP-hard according to Du et al. (1991) even if jobs are allowed for pre-emption.

However, they have concluded that the problem becomes solvable within polynomial time if all jobs have identical processing time and the number of machines does not go beyond two.

Previous works done on identical parallel machines are summarized in Table 1:

Table 1: Summary of previous works on identical parallel machines scheduling with single resource

Research	***Considered problem***	***Performance measure***	***Proposed approach***
Phillips et al. 1998	one machine environment, identical and unrelated parallel machines environment with release time constraint and for both pre-emptive and non-pre-emptive jobs	Average completion time	Approximation algorithm
Yalaoui and Chu 2006	$Pm\|r_j\|\sum C_j$	Total completion time	Exact method
Li et al. 2007	P|pj = 1, rj , outtree|∑Cj	Total completion time	Polynomial algorithm
Nessah et al. 2007	Pm/sds, ri/∑Cj	Total completion time	Exact method
Baptiste et al. 2007	P|r j , pmtn|∑Cj	Average flow time	Linear programming
Li and Chu 2009	P|ri|∑Ci	Total completion time	Backward algorithm (BA)
Ahmed and ElMekkawy 2011	P|ri|∑Ci	Total completion time	Modified forward heuristic algorithm(MFHA)
Shim and Kim 2007	$P_m\|\|\sum T_i$	Total tardiness	Branch and bound algorithm
Biskup et al. 2008	Identical parallel machines	Total tardiness	Heuristic

Table 1 (continued)

Shim and Kim 2008	Identical parallel machines with job splitting property	Total tardiness	Branch and bound algorithm
Sabouni et al. 2010	Identical parallel machines with setup times	Total completion time and maximum lateness	Heuristic
ViniCius et al. 2000	Identical parallel machines	Mean tardiness	Tabu search (TS)
Saricicek et al. 2011	Identical parallel machines with job splitting property	Total tardiness	Tabu search (TS)
Min and Cheng 1999	Identical parallel machines	Makespan	Genetic algorithm (GA)
Kashan and Karimi 2009	Identical parallel machines	Makespan	Particle swarm optimization (PSO)
Lee et al. 2006	Identical parallel machines	Makespan	Simulated anneal-ing (SA)

Most of the scheduling problems listed above have either different constraints or different performance measures compared to the problem of parallel machines scheduling considered in this study. Although Baptiste et al. (2007) focus on the average flow time minimization; their results are not comparable as pre-emption of jobs is considered by the authors. Works of Yalaoui and Chu (2006), Li and Chu (2009) and Ahmed and ElMekkawy (2011) are similar in terms of the problem structure (identical parallel machines scheduling with jobs release time) as flow time of a job can be obtained by subtracting the release time from the completion time of the job. Nevertheless, completion time of a job encounters the makespan of that job. Therefore, total completion time is anticipated to be helpful to compute the total flow time. Job scheduling on parallel machines is traditionally solved by exact methods, for example branch and bound (Yalaoui and Chu, 2006; Nessah et al., 2007; Nessah and Chengbin, 2010). An exact method is proposed by Yalaoui and Chu (2006) to solve the scheduling problem of parallel machines with distinct release time. A polynomial lower bound is proposed in this study by relaxing the release dates or by allowing job splitting. Generalized shortest processing time (GSPT) and Shortest remaining processing time (SRPT) are two modified priority rules used in their paper to generate lower bounds while date constraint relaxation and job splitting are allowed respectively. Their proposed branch and bound algorithm is inspired by the branch and bound used in Chu (1992). An initial solution based on the method of Chu (1992) provides an upper bound. The branch and bound algorithm is applied on several test problems. Although the results obtained from their work and optimal solutions are not stated, a deviation between the average upper bound and the average optimal solution is reported to be 3%.

Nessah et al. (2007) consider the parallel machine scheduling problem with sequence dependent setup time and total completion time as minimization objective. To obtain the local optimality some conditions, according the authors, can be considered as priority rules. Then a dominant subset is defined based on the proposed conditions, and a lower bound which can be solved within polynomial time is generated. Afterwards, the dominance property and lower bounds are accumulated in a branch and bound algorithm to solve the problem in polynomial time. According to them the average deviation of the lower bound from the average optimal solution is 2.95%. Although there is a notification regarding the optimal solution and the lower bounds, the computational results however are focused on the time required to achieve the bounds rather than the completion time of the jobs. The reported results of their paper give no information about the completion time. On the other hand, Nessah and Chu (2010) provide a lower bound by allowing job preemption to minimize the weighted completion time within polynomial time. Different release time of jobs with unavailability period constraint is taken under consideration to make a comparison of their results with the one where unavailability and weighted completion time are not considered. The results however, show an improvement of 46% for several cases.

An extensive study is undertaken by Phillips et al. (1998) on one machine problem and also on identical and unrelated parallel machines. They have proposed a constant factor approximation algorithm to minimize total completion time where jobs come over a certain period of time. Their works have come to a conclusion that a pre-emptive identical parallel machine schedule can be converted into non-pre-emptive schedule with a degradation of time. The commonly used SPT rule is combined with a conversional algorithm to obtain an approximation algorithm from $P|r_j, pmtn|\sum C_j$ to $P|r_j|\sum C_j$. As the target of their work is to develop the algorithm, no results regarding the completion time is mentioned in their paper.

Alternatively, heuristic approaches are commonly found in literature to solve scheduling problem of jobs on identical parallel machines. A Backward algorithm is suggested by Li and Chu (2009) for multi-processor scheduling problem with unequal release time. The jobs are sorted out by using LPT-LRD (longest processing time and longest release date) and scheduled backwards to ensure the influence of the head job of the sequence on the objective function. Their results have outperformed the earliest release date (ERD) and the greedy shortest processing time (greedy SPT). Afterwards, Ahmed and ElMekkawy (2011) propose a modified forward heuristic algorithm (MFHA) which proved to be better than the results obtained by Li and Chu (2009). MFHA is primarily based on priority rules to develop the sub-schedule for each machine with an objective to minimize the flow time of the system. This algorithm allows delay schedule depending on release and processing times of all jobs. As MFHA has already come up with better results, therefore the results obtained from the TS algorithm in this thesis are compared with MFHA rather than the backward algorithm (BA).

Identical parallel machines scheduling is studied by Sabouni et al. (2010), Shim and Kim (2007), Shim and Kim (2008) and Biskup et al. (2008) where all jobs are assumed to be ready at the beginning of the scheduling period.

Though heuristic approaches have been used for many scheduling problems, these approaches are not very efficient (Min and Cheng, 1999). Metaheuristic methods, such as genetic algorithm (GA), tabu search (TS), simulated annealing (SA), particle swarm optimization (PSO) have been applied successfully in the field of combinatorial optimization in view of their characteristics like obtaining near optimal solution, high speed and easy realization (Min and Cheng, 1999 ; Saricicek and Celik, 2011 ; Blige et al., 2004 ;

Sabouni et al., 2010 ; Kashan and Karimi, 2009; Franca et al., 1996 and Sang-II et al., 2006). In this study Tabu search (TS) algorithm has been chosen to deal with the considered scheduling problem.

Bilge et al. (2004) schedule a set of independent jobs with sequence dependent setups on a set of uniform parallel machines such that total tardiness is minimized. Jobs have non-identical due dates and release times. Several key components of TS such as candidate list strategies, tabu classifications, static and dynamic tabu tenure and intensification/diversification strategies are investigated.

ViniCius and Denise (2000) and Saricicek and Celik (2011) focus on tardiness of jobs. Basic tabu search with short term memory described by the authors is followed to develop the TS algorithm in this paper. As long as TS application is concerned, intensification of the search space is an important criterion of the algorithm which is not considered in above mentioned papers. Moreover, the algorithms start with one initial solution to compel the search space within a limited area.

In this thesis, two Tabu search approaches are used to deal with the problem of jobs scheduling on identical parallel machines to minimize the total flow time. Initially the basic tabu search featured with multiple initial solutions (MIS) is applied. Later on, an intensification criterion is implemented to improve the quality of results.

2.2 Scheduling Identical Parallel Machines with Dual Resources

In manufacturing environments, resource flexibility has become an important strategic tool. More and more manufacturers have begun to focus on controlling job processing times by allocating scarce and flexible resources to improve system performance (Daniels et al. 1999). Hence, the problem under consideration in this study is similar as the parallel-machines dual-resources (flexible resource) scheduling problem and is proved to be NP-hard by Daniels et al. (1999), heuristics and metaheuristic approaches are developed to minimize the total flow time.

Exact methods and heuristics, such as Bin Packing and List scheduling approach, Lagrangian relaxation approach, Linear programming and Integer programming, are common to deal with scheduling of parallel machines with a flexible or additional resource constraint (Ruiz-Torres et al. 2007, Ventura et al. 2003, Kellerer and Strusevich 2003, Ruiz-Torres et al. 2007, Grigoriev et al. 2005). Kellerer and Strusevich (2003) have developed a heuristic algorithm to solve the parallel dedicated machines employing the group technology approach to minimize the makespan. A polynomial time approximation scheme is presented. Pre-emption may or may not be allowed. It is assumed that some jobs consume an additional resource. There is a single renewable resource such that one unit of the resource is available at any time. Grigoriev et al. (2005) have considered the unrelated parallel machine scheduling problems with the objective of minimizing the makespan. The processing time of a job is dependent on the allocation of a scarce renewable resource. A two-phased LP rounding technique is used to assign resources to jobs and jobs to machines.

Hu and Chaudhry explore several identical parallel machines scheduling problems with additional resource (worker) for a variety of performance measures. Hu applies heuristics for both job and resource allocations whereas Chaudhry improves the results of those similar problems by using a metaheuristic: Genetic Algorithm (GA).

With the target of minimizing the total flow time of identical parallel machines scheduling with worker assignment problem, a heuristic SPT-A_i, based on the shortest processing time (SPT) to assign jobs and a heuristic named largest marginal contribution (LMC) are proposed by Hu (2005). Here A_i is the constant portion of processing time which does not change with the amount of resource allocated. By using the values of A_i a list of jobs is formed to facilitate the job assignment policy. On the other hand, each idle machine is assigned with a worker at the beginning as the number of workers is greater than the number of machines. Afterwards the next worker is assigned to the machine with the largest marginal contribution. Marginal contribution is defined as the amount of time reduced when one more worker is assigned to a machine. However, the GA algorithm of Chaudhry (2010) outperformed Hu's heuristics for the similar problem.

The study is extended by Hu (2006) to investigate the effect of the processing time which varies with the number of workers assigned. The author has made use of the same SPT rule as applied in his previous work (Hu 2005). Although a brief experimentation is carried out, no significant changes are observed.

A performance measure of total tardiness is investigated in the model of identical parallel machines with worker assignment problems by Hu (2004). The proposed heuristic (SES) in this case is the combination of three commonly used dispatching rules. All the jobs are sequenced using the priority rules in an order of SPT, EDD and Slack. LMC is applied to assign the workers as the previous one explained above. Hu (2006) has executed another study with the performance measure of total tardiness to evaluate the efficiency of the proposed heuristic. In his paper the author has made the simulation more complicated than the previous one by increasing the number of jobs, machines and workers.

Chaudhry and Drake (2009) propose a GA approach to minimize the total tardiness of a set of tasks with known processing times for an identical parallel machines scheduling problem where workers are considered as an additional resource. The performance of the algorithm is evaluated by comparing the results of GA algorithm with the results obtained from SES of Hu (2004). A goal of minimizing the makespan is studied by Chaudhry et al. (2010) and Chaudhry and Mahmood (2011). In the former paper the authors propose a genetic algorithm and the results are compared with the results found from the literature (Hu, 2004)) where SPT/L(east)PA and L(argest)PA heuristic approaches are proposed. In the following year, Chaudhry and Mahmood extend the problem of makespan minimization to evaluate the performance of the GA algorithm. Two sets of problems are taken from the literature and compared with the results of the GA algorithm. The results of the GA algorithm give better solutions proving the algorithm an efficient one.

A decomposition heuristic is suggested by Daniels et al. (1999) to generate an approximate solution of the parallel-machine flexible-resource scheduling problem where job assignment is unknown. A largest process time (LPT) priority index is used to assign n jobs to m machines by sequentially assigning jobs to the first available machine in non-increasing order of job processing time. The resource-allocation algorithm begins by allocating the minimum amount of resource to each machine so that each job can be processed in its slowest mode. Afterwards the highest makespan and the machine with the longest completion time are identified. The minimum amount of resource is allocated that would improve the makespan of at least one job of the machine with highest completion time. Later on, a Tabu search heuristic approach is introduced to deal with the problem. Alternative assignments of jobs are generated by considering all pair wise exchanges of jobs on machines. For each job assignments the above mentioned heuristic is applied to determine the resource allocation policy. The paper concludes that the Tabu search heuristic is more effective than the heuristic approach.

All previous work done on parallel machines with dual resources is summarized in Table 2.

Table 2: Summary of literature review on parallel machines scheduling with dual resources

Research	***Considered problem***	***Performance measure***	***Proposed approach***
Ruiz-Torres et al. (2007)	Parallel machines scheduling with dual resources	Minimize the maximum completion time	Heuristic combining list scheduling and bin packing
Ventura et al. (2003)	Parallel machines scheduling with non-common due dates and additional resource constraint	Minimizing earliness-tardiness	Lagrangian relaxation
Kellerer and Strusevich (2003)	Parallel dedicated machines with an additional resource	Minimizing the makespan	Heuristic employing group technology
Ruiz-Torres et al. (2007)	Uniform parallel machines scheduling with flexible resources	Minimize number of tardy jobs	Integer programming
Grigoriev et al. (2005).	Unrelated parallel machine scheduling problem where processing time depends on a scarce renewable resource	Minimizing the makespan	Approximation algorithms
Po-Chieng Hu (2004)	Identical parallel machine with non-pre-emptive jobs with worker assignment problem	Minimizing total tardiness	SES (SPT,EDD,SLACK) and LMC (largest marginal contribution) heuristics
Po-Chieng Hu (2005)	Identical parallel machine with non-pre-emptive jobs with worker assignment problem	Minimizing total flow time	SPT-A_i and LMC heuristics

Table 2 (continued)

Po-Chieng Hu (2006)	Identical parallel machine with non-pre-emptive jobs with worker assignment problem. Further study is executed on tardiness minimization	Minimizing total tardiness	SES (SPT,EDD,Slack) and LMC (largest marginal contribution) heuristics
Po-Chieng Hu (2006)	Identical parallel machine with non-pre-emptive jobs with worker assignment problem. Performance of the heuristic used in Po-Chieng Hu (2005) is further investigated	Minimizing total flow time	SPT-A_i and LMC heuristics
Imran Ali Chaudhry (2010)	Scheduling a set of tasks and worker on identical parallel machines	Minimizing total flow time	Genetic algorithm (GA)
Chaudhry and Drake (2009)	Scheduling a set of tasks and worker on identical parallel machines	Minimizing total tardiness	Genetic algorithm (GA)
Chaudhry et al. (2010)	Scheduling problem of identical parallel machines with worker assignment constraint	Minimization of makespan	Genetic algorithm (GA)

Table 2 (continued)

Chaudhry and Mahmood (2011)	Scheduling a set of tasks and worker on identical parallel machines	Minimize makespan	Genetic algorithm (GA)
Daniels et al. (1999)	Parallel machines scheduling problem in which impact of resource flexibility is explored	Minimization of makespan	Tabu search heuristic

In summary, most of the above mentioned papers have considered the number of workers equal to or greater than the number of machines. Moreover, the job processing time is a function of the allocated number of workers to that job. It is worth mentioning that, in previous literature, it was assumed that the worker is used over the entire processing time of a job. In addition, some of the previous papers did not consider the job release time. Therefore, the considered problem in this thesis is considered unique compared to the pervious literature. Hence, it is difficult to evaluate the proposed approaches against the literature. Consequently, the performance of the developed methods in this research is compared against each others.

Chapter 3

Scheduling Identical Parallel Machines with Release Time Constraint using Single Resource

This chapter considers minimizing the total flow time of scheduling jobs with different release times on identical parallel machines. A mathematical model and Tabu search (TS) algorithm have been developed to solve the problem. Two neighborhood search methods of TS are proposed. Due to the complexity of the considered problem, the mathematical model is used to solve small cases and the results are compared with the results obtained from the TS algorithm. Moreover, the performance of the TS algorithm is evaluated by comparing its results with the best results of benchmark problems reported in the literature. The proposed TS algorithm has led to better quality solutions compared to the best reported results of literature.

3.1 Introduction

Due to its potential applications, parallel machine scheduling is widely used in real life situations and in variety of fields. These fields include manufacturing, healthcare, computer science, and social science. In modern manufacturing industries, many production control situations, such as semiconductor test facilities, automobile painting lines and plastic injection modeling are treated as job scheduling on parallel machines (Kim and Shin 2003). Moreover, a manufacturing cell consists of parallel machines (for example multiple turning machines) to perform similar operations on many parts of a batch simultaneously (Chaudhry and Drake 2009). The emergence of parallel processor in computer technology has boosted up the interest of research in this area during the last few years too (Chaudhry and Drake 2009). In particular, internet data dispatching system works on the same concept of parallel machines/processor (Li et al. 2010). The same idea is applicable for healthcare systems, such as dental, chemotherapy, surgery and radiotherapy where resources of healthcare unit are similar in the manner of structure (Ahmed and ElMekkawy 2011). The model of parallel machine scheduling with job allocation problems and flow time minimization criteria fits with several social science problems (Mokotoff, 2004). Any practical situation that deals with a series of parallel processing units and jobs processed on them can be characterized as a real life application of jobs scheduling on parallel machines (Mokotoff, 2004).

The scheduling problem in this chapter deals with unequal release times and uses the total flow time as the optimization objective. Here, a set of $\boldsymbol{n}$ number jobs J = $\{J_1, J_2, \ldots\ldots J_n\}$ has to be scheduled on a set of $\boldsymbol{m}$ identical parallel machines M = $\{M_1, M_2, \ldots\ldots M_m\}$. Tabu search (TS) approach has been used to optimize this scheduling problem with $\boldsymbol{r_i}$ and $\boldsymbol{P_i}$ as the release and processing times of job $\boldsymbol{i}$ respectively. It is assumed that one machine can process only one job at a certain time. Pre-emption or splitting of jobs is not allowed. A machine completes processing a job without any type of interruption. The problem under consideration is denoted as $\boldsymbol{Pm|r_i|\sum F_i}$ where Pm stands for identical parallel machines, r_i and F_i for release time and flow time of jobs respectively. Flow time encounters the time of a job spent in the system. Waiting time of a job is being reduced by minimizing the flow time as flow time includes the processing time and the waiting time.

Figure 1 describes the relation among flow time, release time, waiting time and completion time.

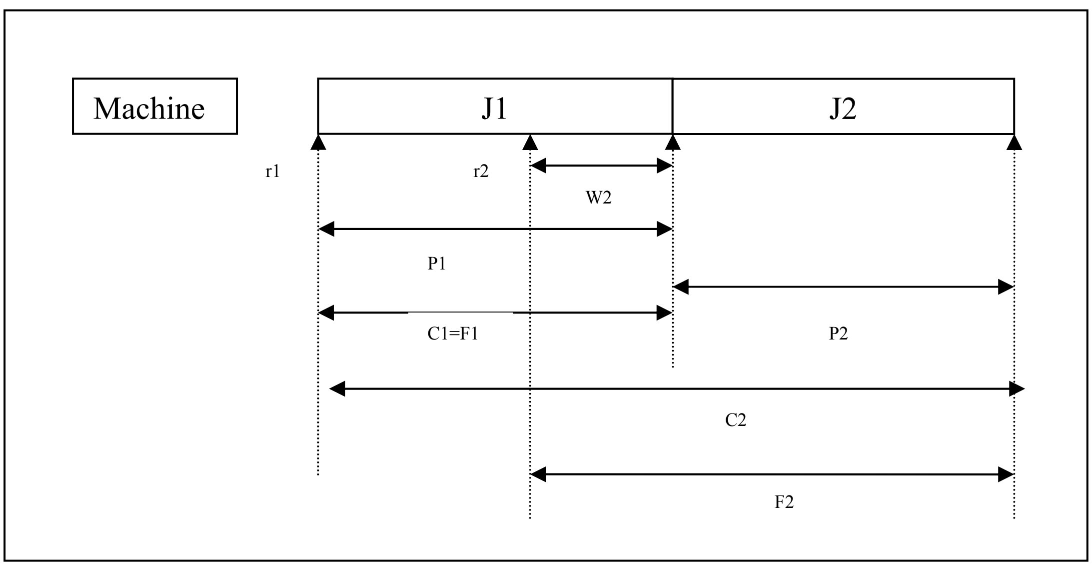

Figure 1: A schematic representation of the correlation among release time, processing time, waiting time, completion time and flow time

According to Figure 1, J1 and J2 are two jobs assigned on a machine where r1 and r2 are their release times respectively. The processing times for J1 and J2 are marked as p1 and p2 respectively. W2 is the waiting time of J2 as it arrives early and has to wait till J1 is finished. Here C1, C2, F1, F2 are the completion times and flow times of job 1 and job 2 respectively. For the first job the completion and the flow time are equal but for the second one flow is obtained by subtracting release time from completion time.

In order to generate an efficient schedule within a reasonable time, Tabu Search (TS) in conjunction with Multiple Initial Solutions (MIS) method is introduced in the algorithm for good solutions. MIS provides a probability of producing initial solutions from different regions of the search space. Thus MIS provides some sort of diversification feature into the algorithm. Later on, for better results an intensification criterion of the algorithm is implemented. This study also suggests a functional equation for the Tabu list size. Moreover, a mathematical model is developed for the problem to obtain optimal results for small cases. The results generated from the TS algorithms are compared with the optimal solutions of small cases solved by the mathematical model. Due to the complexity of the considered scheduling problem, mathematical model cannot obtain optimal solutions for large cases. Therefore, the developed TS algorithms have been used to obtain good quality schedules for large problems and the obtained results are compared with the best reported results of solving the problem in literature. Additional experiments are performed to understand the effect of the used function to generate the release times of the test cases on the total flow time. Section 3.2 of this chapter describes the problem definition and the developed mathematical model followed by Section 3.3 that introduces the TS algorithm. Computational results and discussions are summarized in Section 3.4. The chapter concludes with the scope of future research in Section 3.5.

3.2 Problem definition and mathematical model

The addressed problem consists of scheduling **n** nonpre-emptive jobs $J_1, J_2, \ldots, J_n$ to **m** identical parallel machines to minimize the total flow time, $\sum \mathbf{F_i}$. The processing times of the jobs and their release times are given as $\mathbf{p_i}$ and $\mathbf{r_i}$ respectively. The problem can be denoted as $Pm|r_i|\sum F_i$. The following assumptions are used to develop the mathematical model of the problem:

- Each job has one operation
- Any machine can process any job
- A machine is not allowed to process more than one job at a time
- All the machines are available throughout the scheduling period
- Number of jobs is constant and is greater than the number of machines
- The machine setup time is included in the processing time

The following notations are used in the developed mathematical model:

Parameters:

n	Total number of jobs to be processed
m	Total number of available machines
ri	Release time of job i where $i=1,2,3\ldots\ldots n$
Pi	Processing time of job i where $i = 1,2,3\ldots\ldots n$

Decision Variables:

$C_{[j][h]}$	Completion time of any job at the position h on machine j
$X_{[j][h][i]}$	Binary variable with a value of one if job i is assigned to machine j at position h or zero otherwise

Objective Function:

$$Minimize \sum_{j=1}^{m}\sum_{h=1}^{n} C_{[j][h]} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Constraints:

$$\sum_{j=1}^{m}\sum_{h=1}^{n} X_{[j][h][i]} = 1; \forall i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$\sum_{i=1}^{n} X_{[j][h][i]} = 1; \forall j, \forall h \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$C_{[j][h]} \geq (ri + Pi) * X_{[j][h][i]}; h = 1, \forall i \forall j \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

$$C_{[j][h]} \geq C_{[j][h-1]} + (Pi * X_{[j][h][i]}); h > 1, \forall i \forall j \text{..............................(5)}$$

$$C_{[j][h]} \geq (ri + Pi) * X_{[j][h][i]}; \forall h \forall i \forall j \text{..(6)}$$

$$F_{[j][h]} = C_{[j][h]} - ri; \forall h, \forall i \forall j \text{...(7)}$$

Here, $F_{[j][h]}$ is the flow time of job at the position h on machine j and is obtained by subtracting release time r_i from completion time $C_{[j][h]}$.

Eq. (1) represents the total flow time i.e. the objective function. Constraint (2) ensures that each job is assigned to one machine only, while constraint (3) ensures that at any position of any machine is permitted to process just one job at a time (overlapping of jobs is not allowed). Constraint (4) calculates the job's completion time which must be greater than or equal to the summation of the release time and processing time of the job assigned to the first position (h = 1) of the machine. For all the remaining jobs, the completion time can be calculated using constraint number (5). According to equation (5), the completion time for all remaining jobs must be greater than or equal to the completion time of its predecessor plus its own processing time. Constraint (6) ensures that the completion time of each of the jobs cannot be less than the sum of the time when the jobs arrive and processing time, which confirms that any job is not allowed to be operated on any machine before its release time. If there is no available job, the machine has to wait till the arrival of the first job. Finally, constraint (7) calculates the flow time of each scheduled job.

3.3 Tabu search algorithm

In 1989, Fred Glover developed Tabu Search for combinatorial optimization problems (Glover and Laguna, 1997). The features of the developed TS algorithm for this problem are explained as follows:

3.3.1 Description of the basic TS algorithm

This part outlines the basic TS algorithm adapted to deal with the considered parallel machines scheduling problem by discussing several of the key concepts such as initial solutions, neighborhood search, Tabu tenure and aspiration criterion.

A. Initial solution finding mechanism

An initial solution for the problem must be provided to start the search. Here the initial sequence is generated randomly. Logendran and Subur (2004) showed that the initial solution of TS had influence on the quality of final result. This insight is advantageously used to develop *multiple initial solutions* (MIS). The number of initial solutions has been set to be n/2 for small problems and n/3 for large problems where n is the number of jobs.

For better understanding of the features of TS algorithm, examples are represented in Figure 2 and Figure 3.

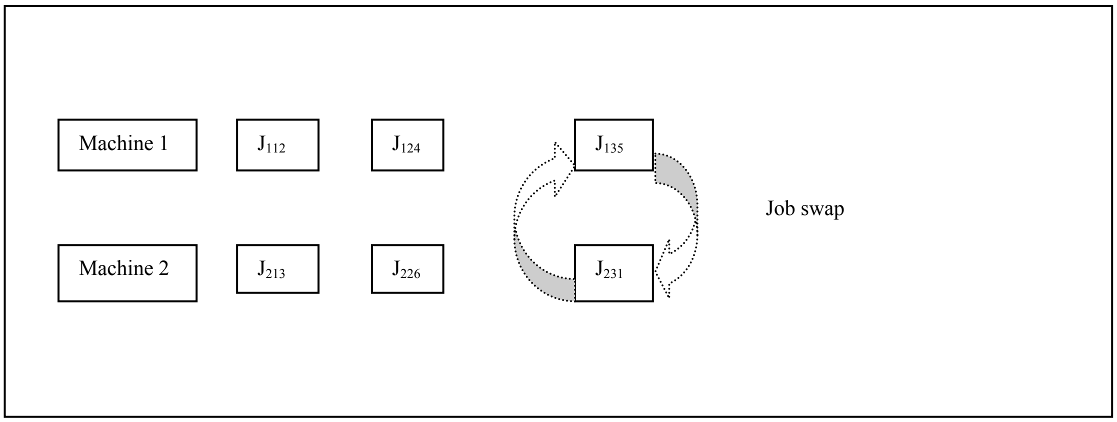

Figure 2: A schematic representation of swap move

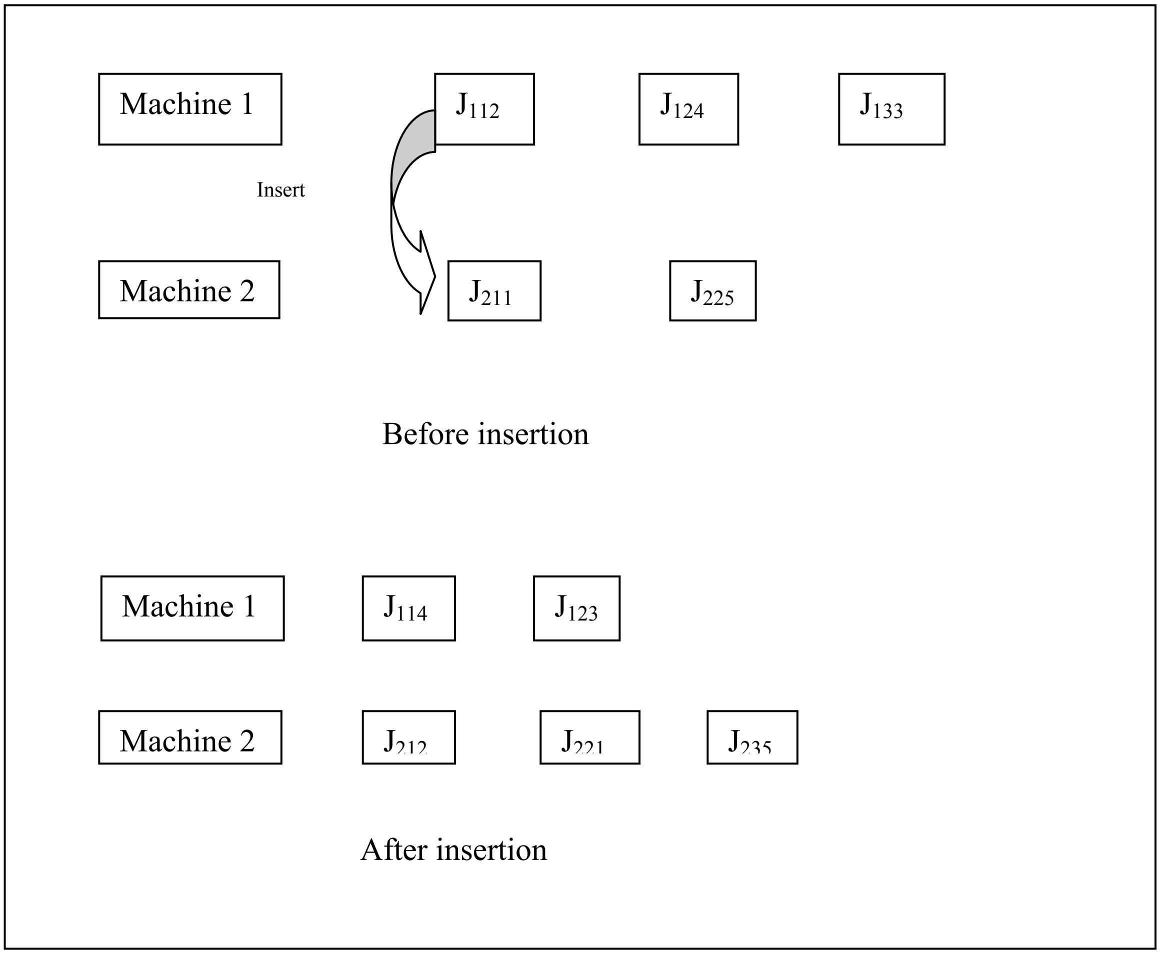

Figure 3: A schematic representation of insert move

Two different neighborhood structures, swap and insert moves, are represented by the figures above. Both the figures have two machines with some jobs represented as J_{jhi} where *j, h* and *i* are marked as machine number, position on the machine *j*, and job number, respectively.

B. Neighborhood structure

The initial solution of TS provides the seed for the algorithm to go forward with the use of neighborhood search process. Neighborhood search is executed using two moves: swap and insert. A swap move selects two jobs either on same machine or on different machines and exchanges their positions. On the other hand, the insert move selects a job randomly and then inserts it to any machine's list of assigned jobs. This study uses the inter machine swaps; i.e. jobs of two different machines are considered for swapping. For example, J_{135} (job number 5 located at the 3rd position on machine 1) and J_{231} (job number 1 located at the 3rd position on machine 2) in Figure 2 exchange their positions on machines which is termed as an inter machine swap. Furthermore, in Figure 3, taking J_{112} and placing it at the first position of machine 2 by pushing the first job J_{211} of machine 2 to the second position, can be termed as job insertion. The arrangements of jobs on machines before and after insertion are specified in Figure 3.

C. Tabu tenure

Tabu tenure is an important feature of TS. It is the length up to which a certain move is not allowed to be performed. In this study, a fixed Tabu tenure approach is employed throughout the search. A range of 5-15 for Tabu tenure is found to be providing good quality results from literature (Glover, 1997; Saricicek and Celik, 2011; Kim and Shin, 2003). Extensive experiments are conducted with varying tabu tenure to achieve good solutions for each cases considered in the experiment of this research. The tabu tenure value is recorded for which best result is obtained for a certain case. Then using the trial and error method a formula is developed depending on the problem size to yield an effective tenure for each class of problems. As a result, the following formula turns out reasonably in the Tabu search algorithm:

Tabu tenure, $TS = m * \sqrt{(n/m)} - \{0.2 * (m-1.5) * n/m\}$

Here, ***m*** is the number of machines and ***n*** is number of jobs.

D. Aspiration criterion

The commonly used condition is considered here for aspiration criterion; a move is permitted to give better solution than the best incumbent solution even if it is Tabu.

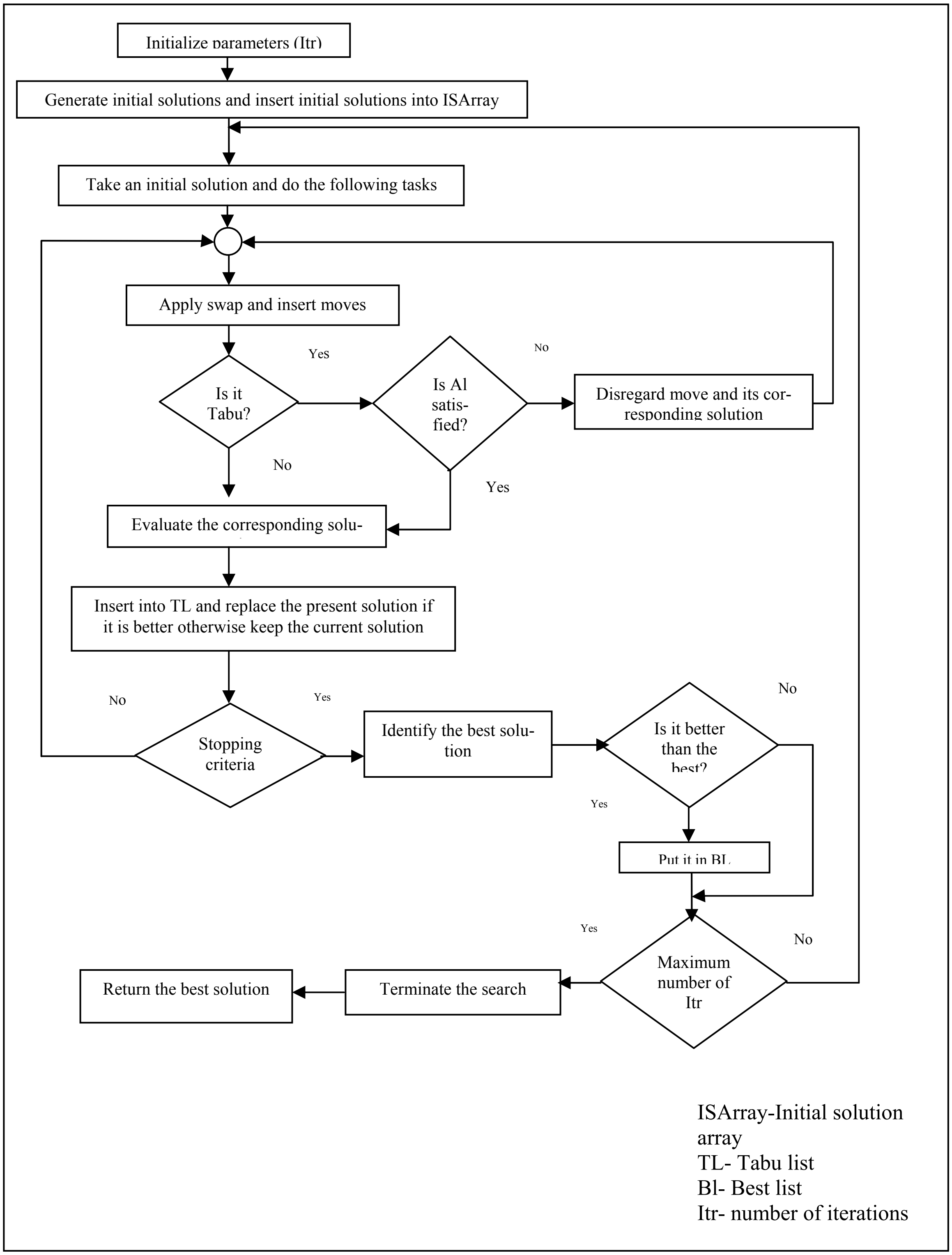

Figure 4: Flow chart of basic TS algorithm

Figure 4 describes the TS algorithm. Initially the parameters of the algorithm, such as number of iteration and number of initial solutions, are determined randomly and inserted in an array named ISArray. For each of the initial solutions swap and insert moves are performed and the performance measure is evaluated. A non- Tabu move is recorded by inserting the job and machine number into a list termed as Tabu List (TL) and the parent solution is replaced if the solution obtained after swap or insert is better than its parent solution. However, if the move under consideration is a tabu, aspiration criterion is checked and a move is accepted only if it satisfies the aspiration criterion. That moves are Tabu but unable to satisfy the condition of aspiration are disregarded for the search continues in the next move. Solutions are sorted to identify the best one with the lowest flow time. During the search, the best solution is recorded in a best list (BL). The best incumbent solution of BL is then compared with the best one out of the swap or inserted moves. Whenever a better solution is found the best list is updated. The best result is obtained after a predetermined number of iterations.

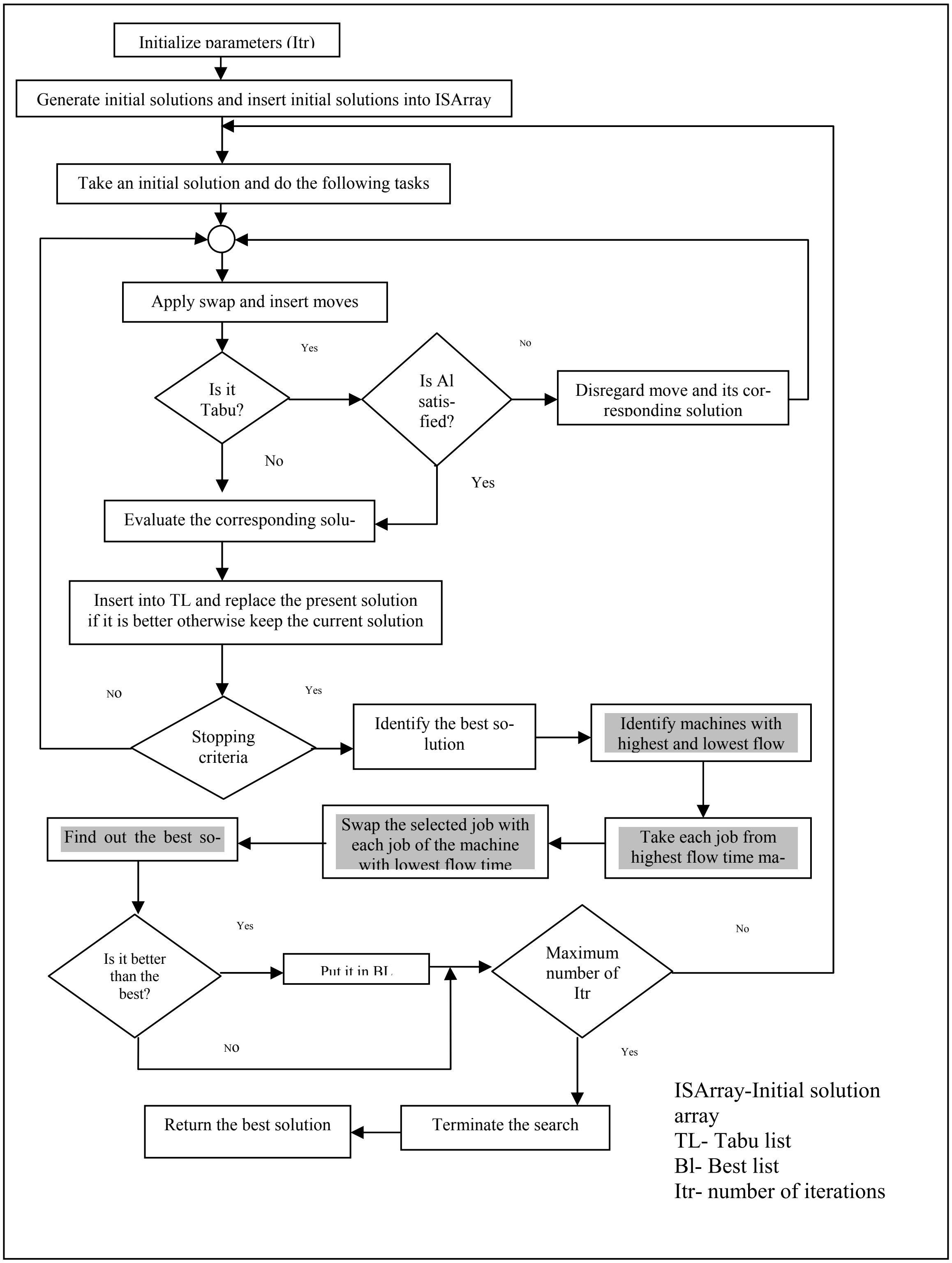

Figure 5: Flow Chart of TS Algorithm with further intensification

3.3.2 Further intensification

Additionally intensification is performed in the TS algorithm to explore the neighborhood for a better solution. The proposed TS algorithm is featured with the property of intensification and is termed as *TS+I* for the rest of the paper. In order to do that, the machines having the highest flow time and lowest flow time are identified for the best solution obtained from swap and insert moves. All jobs of the machine that encounters the highest flow time are swapped with the jobs of the machine with the lowest flow time. The performance measure is evaluated each time. The best solution found after this new swap is then compared with the best incumbent solution of BL as shown in Figure 5. The best one is then recorded in BL. Whenever a stopping criteria occurs the best ever solution obtained from BL is reported as the final result. The steps that differentiate the basic TS algorithm from TS+I are highlighted in Figure 5.

Another conception of multiple initial solutions, instead of an initial solution from the traditional basic TS algorithm, makes it different from other TS algorithms found in literature. Here multiple initial solutions (MIS) provide the essence of diversification because it allows the search engine to go beyond the boundary of any particular solution and to start searching in a completely new area. Tabu search algorithm illustrated here is able to achieve good quality results as MIS is implemented to increase the probability of getting different solutions from different areas of the search space.

3.4 Computational results

Thirty different test cases are generated to evaluate the efficiency of the developed Tabu search (TS) algorithms. Out of these 30 cases, first 10 cases are marked as small problems and the rest are considered to be large based on their size. The small cases consist of number of jobs, n = 10, 20, 50, 100 and number of machines, m = 2, 5, 10. Test cases with number of jobs, n = 200, 300, 400, 500 and number of machines, m = 10, 20, 30, 40, 50 are the large ones.

The TS algorithm is coded with Microsoft Visual Studio 2010 C++. The mathematical model is solved using IBM ILOG CPLEX 12.3. The experiments are conducted on a Pentium® 4 with 3.00 GHz CPU and 1 GB RAM.

The parameters of the TS algorithm such as number of iterations and number of initial solutions are set based on performing detailed experiments using several common values found from literature. The number of iteration varies from 100 to 5000, tabu list follows the equation $m * \sqrt{(n/m)} - \{0.2 * (m-1.5) * n/m\}$, where, $\boldsymbol{m}$ is the number of machines and $\boldsymbol{n}$ is number of jobs. The number of initial solutions is set to n/2 or n/3 for small and large problem sizes respectively. Release times and processing times are integer values generated within the interval of [0,100] and [1,100] respectively with uniform distribution following the literature (Li and Chu, 2009; Ahmed and ElMekkawy, 2011). In section 3.4.1, further experiments are analyzed for the effect of selected values of the release time on the system's performance.

For each solved problem, the code is run for 20 times and the best total flow time is chosen for comparison. Due to the complexity of the considered problem (NP-hard), only small cases can be solved using the mathematical model. For this reason small cases are compared with the optimum solutions and large cases are compared with the ones found in Ahmed and ElMekkawy (2011). Appropriate results to compare with, are found only in the work done by Ahmed and ElMekkawy (2011). The literature review shows that Yalaoui and Chu (2006) and Phillips et al. (1998) dealt with the same scheduling problem. However, the detailed results of flow times or completion times are not given by Yalaoui and Chu (2006) and Phillips et al. (1998).

The difference (or the gap) of total flow time between TS and the optimum value or the values from the Modified Forward Heuristic Algorithm (MFHA) are calculated using the following equations.

*Diff1 (%) = {(Proposed TS – Optimum solution)/ Optimum solution}*100…. (8)*

*Diff2 (%) = {(Proposed TS – MFHA)/ MFHA}*100…………………………… (9)*

Table 3: Computational results for small problems

Case No.	n	m	Values of flow time				Diff2 (TS+I Versus MFHA)	Diff2 (TS Versus MFHA)	Diff1 (TS+I Versus Optimum)	Diff1 (TS Versus Optimum)
			TS +I	TS	MFHA	Optimum				
1	10	2	49	51	54	49	-9.25	-5.55	0	4.08
2	20	2	143	151	170	140	-15.88	-11.17	2.14	7.85
3	50	2	516	547	550	498	-6.18	-0.55	3.62	9.83
4	100	2	1072	1077	1099	1002	-2.45	-2.00	6.98	7.48
5	10	5	59	59	59	59	0	0	0	0
6	20	5	77	79	82	71	-6.09	-3.65	8.45	11.26
7	50	5	252	261	272	244	-7.35	-4.05	3.27	6.96
8	100	5	426	441	449	423	-5.12	-1.78	0.709	4.25
9	50	10	119	121	132	114	-9.84	-8.33	4.38	6.14
10	100	10	216	219	228	205	-5.26	-3.95	5.36	6.82
Average							-6.75	-4.11	3.49	6.47

* n and m are Number of jobs and Number of machines respectively
* TS: Basic TS without intensification
* TS+I: TS with further intensification

Table 4: Computational results for large problems

Case No.	n	m	Values of flow time			Diff2 (TS+I Versus MFHA)	Diff2 (TS Versus MFHA)
			TS +I	TS	MFHA		
1	200	10	510	523	531	-4.08	-1.63
2	300	10	743	762	791	-6.12	-3.72
3	400	10	988	1012	1056	-6.47	-4.20
4	500	10	1276	1287	1319	-3.30	-2.43
5	200	20	249	258	268	-7.38	-4.03
6	300	20	375	388	396	-5.43	-2.15
7	400	20	497	517	527	-5.81	-2.02
8	500	20	623	635	659	-5.51	-3.69
9	200	30	172	178	182	-5.85	-2.56
10	300	30	241	250	266	-9.55	-6.18
11	400	30	335	343	353	-5.17	-2.91
12	500	30	415	426	439	-5.66	-3.16
13	200	40	129	134	140	-8.32	-4.77
14	300	40	201	209	215	-6.72	-3.01
15	400	40	266	271	276	-3.93	-2.92
16	500	40	319	328	338	-5.63	-2.97
17	200	50	109	111	116	-6.50	-4.08
18	300	50	152	157	164	-7.69	-4.65
19	400	50	201	208	215	-6.66	-3.41
20	500	50	253	257	266	-4.92	-3.41
Average						-6.04	-3.98

* n and m are Number of jobs and Number of machines respectively
* TS: Basic TS without intensification
* TS+I: TS with further intensification

Table 5: Time required for TS algorithms and optimal solution

Case No.	n	m	Time required (Sec)			
			TS+I	TS	MFHA	Mathematical model
1	10	2	1.2	1.15	1	1001
2	20	2	1.7	1.2	1.2	2294
3	50	2	2	1.8	1.3	5466
4	100	2	2.8	1.9	1.6	12520
5	10	5	1.9	1.95	1.4	1890
6	20	5	5.9	2.85	2.5	2626
7	50	5	3.9	2.95	2.2	4260
8	100	5	6.9	5.8	3.2	16392
9	50	10	8.2	7.7	3.5	4235
10	100	10	11.6	7.8	3.9	12597
Average			4.61	3.51	2.18	6328.10

* n and m are Number of jobs and Number of machines respectively
* TS: Basic TS without intensification
* TS+I: TS with further intensification

It can be noticed from Table 3 that TS+I obtained optimum solutions of two cases whereas the TS obtained the optimum of one case. The average difference from the optimum solution is 3.49% and 6.47% for TS+I and TS respectively. An improvement of 6.75% and 4.11% is achieved with respect to MFHA.

Results of large sized problems are summarized in Table 4. The proposed algorithms of Tabu search (TS and TS+I) are compared with the MFHA of Ahmed and Elmekkawy (2011). Both algorithms are proved to be better than MFHA in terms of quality of solution as the performance measure is the flow time here. However, the TS+I outperformed the TS algorithm.

The solution CPU time is reported in Table 5. The average CPU times of the TS, TS+I and MFHA are 3.51, 4.16 and 2.18 seconds, respectively. On the contrary, CPLEX took 6328.10 seconds on average to find the optimal solution.

3.4.1 Effect of varying release time on flow time

In addition to the experiments of generating the release times based on the used procedure by Ahmed and ElMekkawy (2011) and Li and Chu (2009), another experiment is performed to identify the effect of release times on the system's performance. Cheng et al. (2002) generated the release time using the uniform distribution with different ranges: [0, 100], [0, 500], [0, 1000], [0, 2500], and [0, 5000]. Nessah et al. (2007) used a function of the number of jobs and machines to generate the release time. This function is Uniform [0, 50.5*N*α/M], where N and M are number of jobs and machines, respectively and α is a parameter that can take a value of 0.6, 0.8, 1.5, 2, or 3. Distributing the arrival of jobs throughout the makespan of the schedule makes the problem more realistic. In this study, a function is developed to generate release times of jobs. This function is R [0, $(n/m)*P_{avg}*\alpha$] which is slightly different from the proposed function of Nessah et al. (2007). The parameters *n* and *m* are number of jobs and number of machines, respectively. P_{avg} is the average processing time of all jobs. The parameter α is defined as {0.25, 0.5, 0.75, 1}. It is assumed that the term $(n/m)*P_{avg}$ provides an approximate idea about the makespan to allow jobs to arrive in a controlled manner. It is assumed that all the jobs are equally distributed among the machines (n/m). Makespan of a machine is then estimated by multiplying the number of jobs on a machine by the average processing time of jobs P_{avg}. Four different values of the parameter α distribute the arrival of jobs such that all jobs are ready within the first, second, third and fourth quarter of the estimated makespan to generate four different arrival scenarios.

First 10 cases (small sized problems) are executed with the new range of release times using the same code of TS+I as it has proved better than the TS. Results of the four ranges are summarized in Table 6 below.

Table 6: Results of TS+I with varying release time

Case No.	n	m	Values of flow time With R (0, α*n/m*P_{avg})			
			α=0.25	α=0.5	α=0.75	α=1.00
1	10	2	118	71	82	45
2	20	2	157	172	149	127
3	50	2	483	471	402	547
4	100	2	1024	1045	1011	1118
5	10	5	56	57	49	57
6	20	5	84	68	96	93
7	50	5	231	195	198	219
8	100	5	509	419	465	537
9	50	10	122	101	114	109
10	100	10	212	186	233	236

* n and m are Number of jobs and Number of machines respectively
* TS+I: TS with further intensification

The experiments show the effect of job arrival pattern on the flow time. For each of the cases best results are obtained in different regions of release time depending on the problem size. It is observed that α=0.5 gives better results in 50% of the cases. In order to get good quality results for small sized problems like 10 jobs on 2 machines and 20 jobs on 2 machines, the jobs are distributed throughout the makespan. A value of α=0.75 proves suitable for the rest of 3 cases which are middle sized problems. This experiment proves that job arrival pattern is an important parameter to study. Jobs have to wait for a long time if they arrive at the beginning and thus increase the total flow time. On the other hand, if jobs are controlled to come to the system throughout the time span of operation, it encounters less waiting and consequently less total flow time.

Moreover, the results obtained from TS+I for different ranges of release time are compared with the results of MFHA. MFHA code is slightly modified to accommodate the changes of release times according to the function Uniform [0, (n/m)*P_{avg}*α]. Table 7 exhibits the comparison for all four values of α.

Table 7: Comparison of TS + I & MFHA with varying release time

Case No.	n	m	Values of flow time With R (0, α*n/m*P_{avg})					
			α=0.25			α=0.5		
			TS+I	MFHA	Diff3 (%)	TS+I	MFHA	Diff3 (%)
1	10	2	118	127	-7.1	71	76	-6.6
2	20	2	157	169	-7.1	172	185	-7.0
3	50	2	483	493	-2.0	471	481	-2.1
4	100	2	1024	1039	-1.5	1045	1098	-4.8
5	10	5	56	57	-1.8	57	58	-1.7
6	20	5	84	91	-7.7	68	73	-6.8
7	50	5	231	248	-6.9	195	207	-5.8
8	100	5	509	535	-4.9	419	438	-4.3
9	50	10	122	131	-6.8	101	107	-5.6
10	100	10	212	229	-7.4	186	203	-8.4
Average					-5.3			-5.3

Table 7 (continued)

Case No.	n	m	Values of flow time With R (0, α*n/m*P_{avg})					
			α=0.75			α=1.00		
			TS+I	MFHA	Diff3 (%)	TS+I	MFHA	Diff3 (%)
1	10	2	82	87	-5.7	45	49	-8.2
2	20	2	149	163	-8.6	127	135	-5.9
3	50	2	402	416	-3.4	547	568	-3.7
4	100	2	1011	1023	-1.2	1118	1142	-2.1
5	10	5	49	50	-2.0	57	58	-1.7
6	20	5	96	105	-8.6	93	99	-6.1
7	50	5	198	211	-6.2	219	231	-5.2
8	100	5	465	495	-6.1	537	568	-5.5
9	50	10	114	121	-5.8	109	115	-5.2
10	100	10	233	255	-8.6	236	246	-4.1
Average					-5.6			-4.8

* n and m are Number of jobs and Number of machines respectively
* TS+I: TS with further intensification

In Table 7, the difference is defined as:

*Diff 3 (%) = {(TS+I – MFHA)/MFHA}*100………………….. (10)*

The data of Table 7 show the dominance of TS+I over MFHA with an average of 5% improvement in results. TS+I provides improved results (5.3%) in comparison to MFHA for both $\alpha = 0.25$ and $\alpha = 0.5$. However, the improvement rises to 5.6% where the value of α is 0.75, whereas the lowest improvement is observed if the jobs arrive till the end of the makespan ($\alpha = 1$).

An example of the template generated with the help of TS+I algorithm is presented in Figure 6 where the number of jobs and machines are 10 and 5 respectively. The machines remain idle at the beginning of the schedule due to unavailability of jobs during that time. The flow time is 59 for the system. Afterwards, jobs are operated one after another as all the jobs arrived by the time first lot of jobs of each machines had been completed. On the other hand, when release times of jobs are generated following the function $[0, (n/m)*P_{avg}*\alpha]$ and using 0.75 as the value of α, the template improves with a flow time of 49 which is shown in Figure 7. It is due to the control on job arrival through the function. The arrival of jobs is distributed in such a manner that reduces the flow time of jobs.

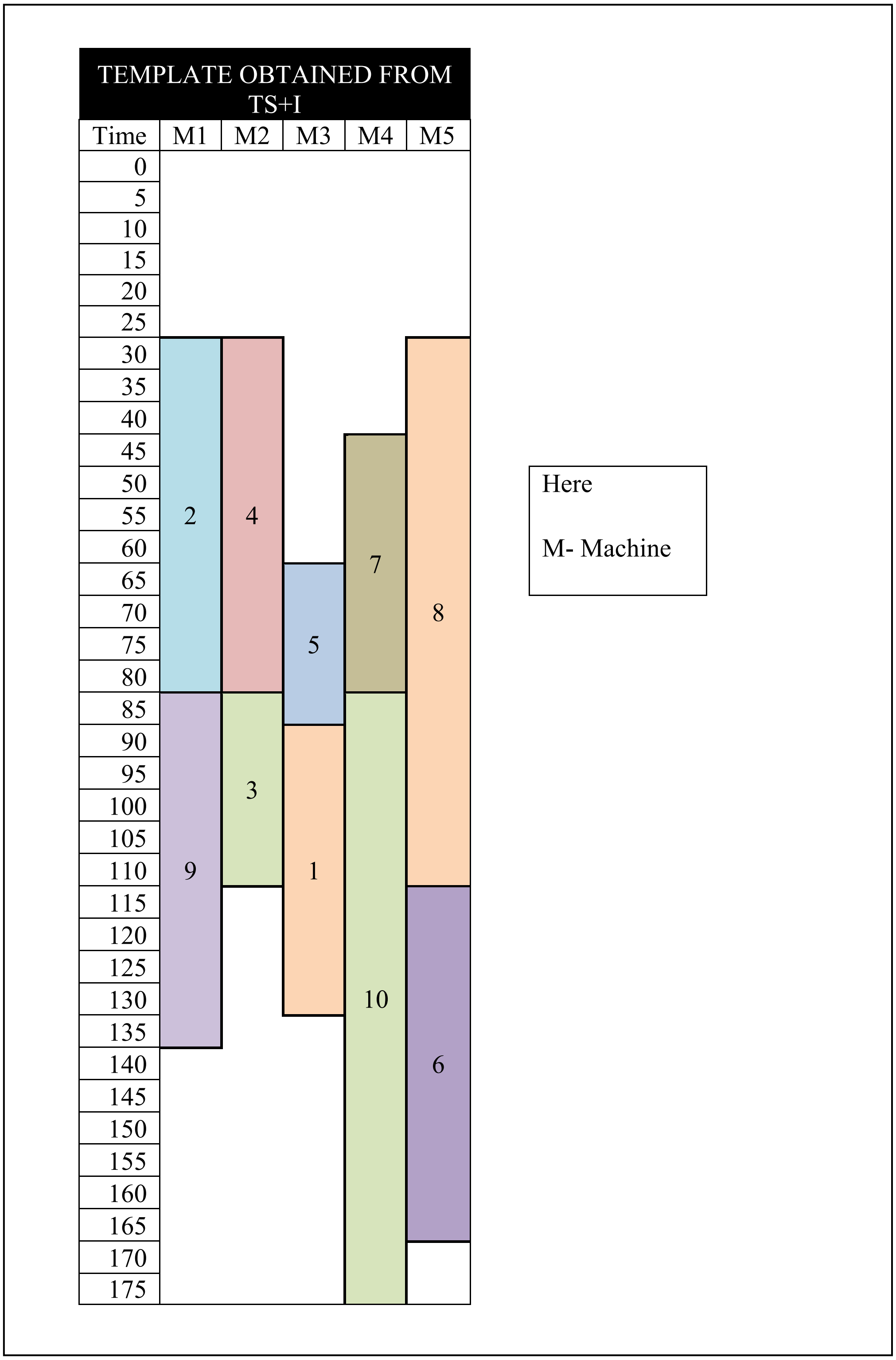

Figure 6: Template obtained from TS+I

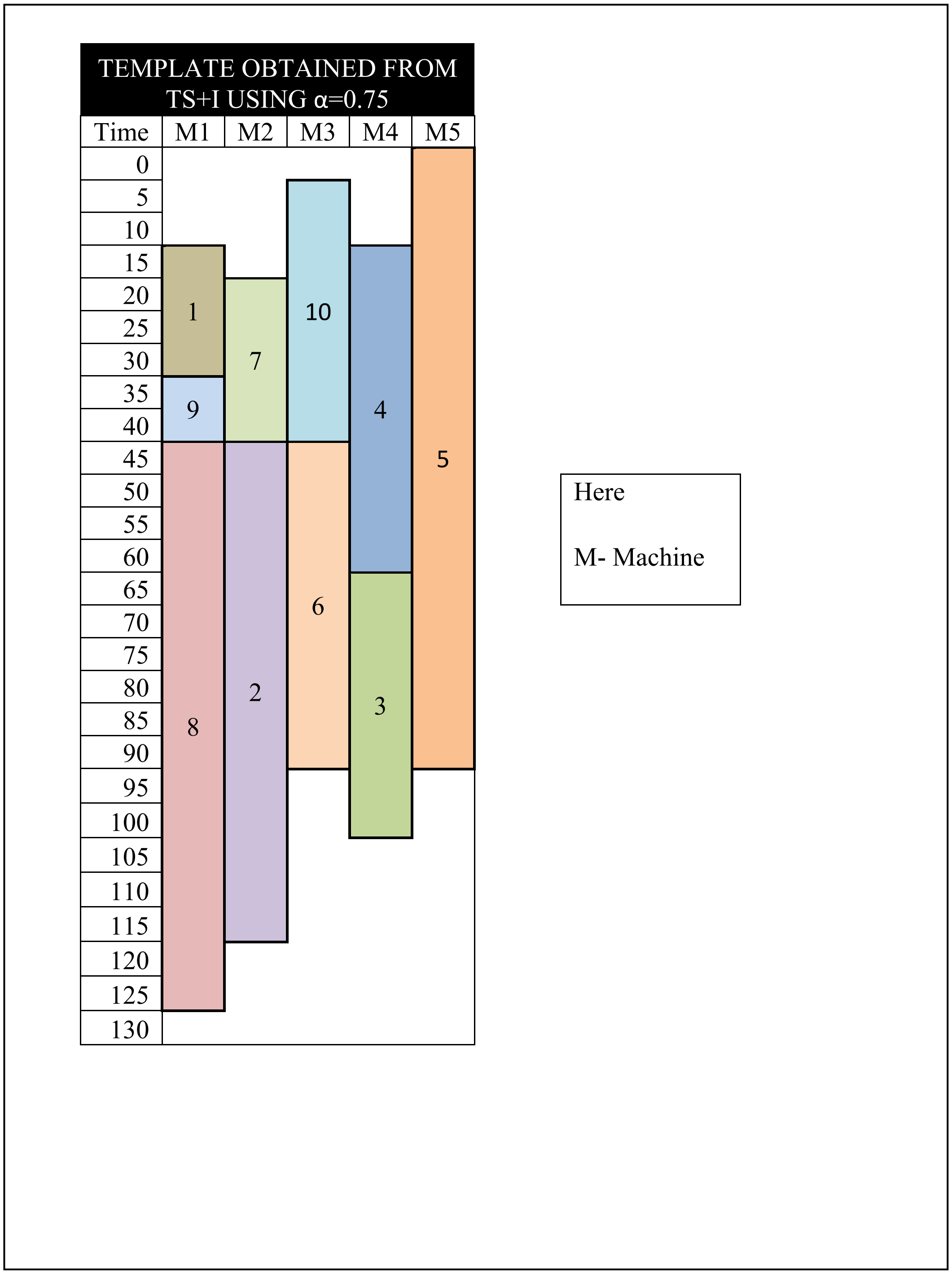

Figure 7: An example of template obtained from TS+I using α=0.75

3.5 Conclusion

Identical parallel machines scheduling problem with different jobs release times is investigated in this chapter. A Tabu search algorithm with multiple initial solutions (MIS) and intensification is developed. In addition, a mathematical model is developed to obtain optimum solutions of small sized problems and compare with the results of the developed TS algorithm. Furthermore, larger test cases are solved by the TS algorithm and compared with the best results of the literature. The proposed TS algorithm leads to better quality of solutions than the reported results in the literature. Additional experiments are executed for jobs release times to make the model more realistic. The release time of job is represented as a function of number of jobs and number of machines to change the arrival pattern of jobs. The experiment shows that the flow time can be improved by distributing the arrival of jobs instead of concentrating them at the beginning of the scheduling time period. As Tabu search has proved better solutions than the results obtained from literature, the algorithm of TS has good potential to be used in several manufacturing and healthcare environments. Although the algorithms reported in this chapter are not able to generate optimum solutions for all cases, in future more advanced features like diversification and dynamic Tabu tenure can be included to improve the efficiency of the algorithm. Other meta-heuristics like Simulated Annealing, Particle Swarm Optimization and Genetic Algorithm can also be investigated in conjunction with Tabu Search to get a hybridized algorithm to achieve better results.

Chapter 4

Scheduling Identical Parallel Machines with Release Time Constraint using Dual Resources

This chapter is aimed at developing an efficient scheduling template with the help of a metaheuristic method: Tabu Search (TS), for a chemotherapy unit in a healthcare clinic with dual resources. Here patients have their own arrival time in the clinic and not all patients require same times for their treatment. With the target of minimizing the total flow time of patients within the system of chemotherapy clinic, the Tabu search algorithm for dual resources (TSD) has been developed. In this case, Tabu search algorithm has been implemented with Multiple Initial Solutions (MIS). In addition, an intensification criterion of TS has also been applied here which is similar to the algorithm described in Chapter 3. Moreover, a simulation study is carried out by developing different heuristics using the existing dispatching rules. Initially, Tabu search algorithm is combined with the best reported heuristic to develop Tabu search algorithm with heuristic (TSHu). Eventually, an extended experiment is done and the performance of the TSD algorithm is evaluated by comparing its results with those obtained from heuristics and TSHu. The proposed algorithm has led to a better solution than other approaches. Finally, a template is proposed for the chemotherapy treatment unit to schedule patients.

4.1 Introduction

A chemotherapy treatment unit serves patients with different disease groups. Among all the resources used in the treatment unit, chairs and nurses are considered to be the imperative ones as the schedule for the patients mostly depends on the availability of these two resources. Here, all the chairs are similar and patient's treatment time remains the same regardless of the assigned chair to provide the treatment. On the other hand, availability of nurses has to be considered for loading and unloading patients. In other words, loading and unloading of patients represent the need of each patient for a nurse to prepare and discharge him/her at the start of the treatment and the end of the treatment respectively. Therefore, the problem can be formulated as a dual resources' scheduling problem with W workers and m parallel identical machines, where $W<m$. Moreover, unequal release time and non-common processing time (treatment time) of patients make the situation similar to the identical parallel machines (Pm) problem with the release time (r_i) constraint where worker (nurse) is considered as an additional resource. Thus the problem can be denoted as $Pm,W \mid r_i \mid \sum F_i$ where, F_i represents the flow time of jobs. Flow time encounters the time of a job (patient) spent in the system. The relations among flow time, release time, waiting time and completion time are already described in Section 3.1 with the use of Figure 1.

It is known that the release time is the time when a patient is expected to arrive. The release time constraint is used because of the following reasons:

i) Some patients need to visit Lab area or the Physician prior to visiting the chemotherapy treatment. Therefore, it is reasonable to consider this requirement before booking a patient.
ii) Most of the chemotherapy drugs are made at the day of the treatment, as it is not cost efficient to preserve those drugs. The pharmacists have to work all over the day to prepare those drugs. Very few drugs are being prepared the day before the treatment, mostly for the longer treatment such as the twelve-hour treatment. Consequently the chemotherapy preparation time can also be referred as release time constraint.
iii) There are patients with different treatment durations that need nurses at the beginning and end of their treatments. Moreover, the treatment centre employs different number of nurses with different shift schedule over the working day. Therefore, matching the arrival pattern of patients with the nurses' availability could improve the system performance and simplify the scheduling problem. This issue has been addressed in a simulation study by Ahmed (2011).

However, experiments are conducted in this research to generate heuristics for the problem of dual resources using several dispatching rules. Later on, a Tabu search algorithm is developed to generate a sequence of patients on chairs which eventually accommodates a heuristic to assign the second resource. Then the algorithm is termed as Tabu search algorithm with heuristic (TSHu). Finally, modifications are done on the Tabu search algorithm to fit both the resources simultaneously. The algorithm is then addressed as Tabu search for dual resources (TSD). At the end a template is developed for the overall chemotherapy treatment unit.

The rest of the chapter is arranged as: Section 4.2 describes the problem followed by Section 4.3 where the heuristics, TSHu and TSD for the dual resources are introduced. Computational results and discussions are summarized in Section 4.4. A case study is conducted and presented in Section 4.5. The chapter concludes with the scope of future research in Section 4.6.

4.2 Problem description

The addressed problem consists of scheduling **n** nonpre-emptive jobs (patients) $J_1, J_2, \ldots,$ J_n to **m** identical parallel machines (chairs) to minimize the total flow time, $\sum \mathbf{F_i}$, where W workers (nurses) are needed to aid the jobs. The processing times of the jobs and their release times are given as $\mathbf{p_i}$ and $\mathbf{r_i}$ respectively. The problem can be denoted as $\boldsymbol{Pm, W \,|r_i|\, \sum F_i}$. The following assumptions are made prior to the development of the algorithms.

- Each job has one operation
- Any machine can process any job
- A machine is not allowed to process more than one job at a time
- All the machines are available throughout the scheduling period

- Number of jobs is constant and is greater than the number of machines
- The number of workers is less than the number of jobs and machines; $W<m$
- The time needed for a worker to load or unload a job is a fixed amount of 15 minutes.
- Only one worker is required for loading or unloading of a job.

As stated earlier, a decision has to be taken by a worker either to load a new job (from among waiting jobs) on an available machine or unload a job that finished the processing. Hence, the loading and unloading has to be done by a worker, both tasks (loading/unloading) have to wait for an available worker and an available machine in case of loading a job. Accordingly, a decision has to be taken to load or unload a job at one of the following mutually exclusive states of the system:

State A: Only a set of jobs waiting for loading on a machine.

State B: Only a set of jobs waiting for unloading from a machine and leave the system.

State C: A set of jobs waiting for loading and another set of jobs waiting for unloading at the same time.

At the very beginning, all the machines and the workers are free and a set of jobs are waiting to be loaded. Any of the workers can take a job to start the operation. The decision of which job to choose would be taken by the worker based on either of the states of jobs mentioned above. Thus the time of availability of worker, machine and job change the state of the system. To facilitate the decision and to schedule jobs, simple dispatching rules can be incorporated for each of the states. The usage of the dispatching rules will be discussed in Section 4.3.

4.3 Solution approaches

4.3.1 Heuristics

The heuristics described in this section is designed to generate approximate solutions for the identical parallel machines scheduling problem with dual resources to minimize the total flow time of the system. Whenever any of the three situations explained in Section 4.2 (State A, B or C) arise, a worker has to make a decision. The rules to be followed by a worker depend on the state of the system. An example of a heuristic combination is described below:

State A: Take the job with largest processing time (LPT).

State B: A worker would choose a job that has completed the machining process earlier than another job for unloading.

State C: A worker would choose a job to be loaded on a machine if the job has its Critical Ratio (CR) < 1. Otherwise the worker would unload the job.

It is because a job with CR<1 is falling behind the schedule. So the job has to be loaded on a machine. The value of CR indicates the time left for a job to be processed with respect to its processing time.

CR > 1: the job is ahead of schedule and has some slack time

CR = 1: the job is on time

CR < 1: the job is behind the schedule

Here, CR is calculated using the following equation:

CR = {(due date or time – current date or time)/ processing time needed by the job}

Following the concept of the heuristic mentioned above several combinations of dispatching rules can be generated for different states. The combinations for all three states are summarized in Table 15 of Appendix A.

The selected ten combinations are given in Table 8 below:

Table 8: Selected combinations

	State A		State B		State C	
Combination No.	Primary rule	Secondary rule	Primary rule	Secondary rule	Primary rule	Secondary rule
1	LPT	Random	Random	None	Load if CR<1	Otherwise Discharge
2	LPT	Random	FIFO	Random	Load if CR<1	Otherwise Discharge
3	LPT	Random	Random	None	Discharge first	
4	LPT	Random	FIFO	Random	Discharge first	
5	LPT	Random	FIFO	Random	Load if Slack time <=0,	Otherwise Discharge
6	SPT	Random	Random	None	Load if CR<1	Otherwise Discharge
7	SPT	Random	FIFO	Random	Load if CR<1	Otherwise Discharge
8	SPT	Random	Random	None	Discharge first	
9	SPT	Random	FIFO	Random	Discharge first	
10	SPT	Random	FIFO	Random	Load if Slack time <=0,	Otherwise Discharge

Experiments have shown that two combinations provide best results for most of the cases. The one that gives the best results in maximum number of cases is accepted here to combine with the Tabu search algorithm.

4.3.2 Tabu Search Heuristic (TSHu) Algorithm

The Tabu search algorithm developed in Section 3.3 is utilized in this chapter for the dual resources problem. Tabu search algorithm with further intensification (TS+I) is the algorithm which is used instead of the basic TS algorithm of Section 3.3. All the features described in Section 3.3 are true for the TSHu algorithm. The solution obtained from the Tabu search algorithm then follows the heuristic algorithm which is developed based on the dispatching rules of combination 3 of Table 8. A free worker chooses the job with the longest processing time if more than one job is waiting to be loaded at any time. Among the jobs those need only unloading at any time, any of those can be taken by the worker for unloading. In situations where a set of jobs waiting for loading and another set of jobs waiting for unloading at the same time, a free worker is supposed to unload the job first.

4.3.3 Tabu search heuristic for dual resources (TSD)

Unlike TSHu the Tabu search for dual resources (TSD) applies the heuristic during the search of each alternative assignment of jobs on machines i.e. after each iteration the heuristic algorithm is applied to the Tabu search algorithm. Alternative assignments are generated using swap and insert moves described in Section 3.3. For each job assignment, the heuristic algorithm is applied to assign the worker and to determine the corresponding value of the performance measure criterion (total flow time). If an improvement is observed in the total flow time, the best incumbent solution is updated and the move is defined as tabu. After a predetermined number of iteration the best ever value is reported as the final solution.

4.4 Computational results

The proposed three approaches: heuristics, TSHu and TSD are coded using Microsoft Visual Studio 2010 C++. The experiments are conducted on a Pentium® 4 computer with 3.00 GHz CPU and 1 GB RAM. Initially, ten test cases are generated to evaluate the performance of the algorithms. The key design variables included in this study are the jobs, machines and workers. By varying the design variables the test cases are generated, which guides the way to select the best combination out of all ten chosen heuristic combinations of Table 8. The results of test cases with number of jobs, n = 20, 30, 40, 50, 60 and number of machines, m = 5, 6 are considered with just two workers in Table 10. Here, the release time and processing time are integer values generated within the interval of [0,100] and [1,100] respectively with uniform distribution. Later on, another experiment is executed with varying release time.

Table 9: Computational results of heuristics

Case No.	Number of jobs	Number of machines	Number of workers	comb1	comb3	comb5	comb6	comb7	comb8	comb9	comb10
1	20	5	2	6075	5640	6030	6045	6045	5880	5680	6000
2	20	6		4290	4290	4290	4290	4290	4290	4290	4290
3	30	5		13440	11655	11655	12495	12495	11710	11710	12250
4	30	6		9240	8940	9240	10590	10590	9285	9285	9480
5	40	5		18165	19140	18180	20400	20400	18825	18590	20400
6	40	6		16680	16320	17580	16840	17580	16580	16580	16720
7	50	5		30525	32655	30585	39555	39655	31620	31655	39555
8	50	6		24855	25860	24855	25800	25540	25280	25280	25800
9	60	5		67845	55005	67845	55080	55080	65585	65585	65585
10	60	6		50745	43125	43485	60480	60480	47550	47550	45735

From Table 9 it is observed that combination 3 provides the best solutions for 7 cases out of 10. Combination1 on the other hand provides the best solution in several cases. Especially when combination 3 could not achieve the best one and thus combination1 is accepted as the second best. The results of combination 3 and combination 4 are similar.

The literature of identical parallel machines scheduling with single resource in Section 3.4.1 shows that a wide range of release time is more realistic than having the arrival pattern of all the jobs squeezed at the beginning of the schedule horizon (Cheng et al., 2002; Nessah et al., 2007). The same function developed in Section 3.4.1 is followed in this chapter. The function is R [0, $(n/m)*P_{avg}*\alpha$] which is slightly different from the proposed function of Nessah et al. (2007). The parameters *n* and *m* are number of jobs and number of machines, respectively. P_{avg} is the average processing time of all jobs. The parameter α is defined as {0.25, 0.5, 0.75, 1}. It is assumed that the term $(n/m)*P_{avg}$ provides an approximate idea about the makespan and allows jobs to arrive to the system in a controlled manner. It is assumed that all the jobs are equally distributed among the machines (n/m).

Makespan of a machine is then estimated by multiplying the number of jobs on a machine by the average processing time of jobs P_{avg}. Four different values of the parameter α distribute the arrival of jobs such that all jobs are ready within the first, second, third and fourth quarter of the estimated makespan to generate four different arrival scenarios.

Five cases (case no. 1, 3, 5, 7, 9) in Table 9 are evaluated to identify the best fitted value of α through Table 16 to Table 20 in Appendix A.

Table 10: Comparison of two different arrival patterns

Case No.	Number of jobs C1	Number of machines C2	Number of workers C3	Current arrival pattern C4	Changed arrival pattern C5	% difference (C4 vs C5)	α	Combination No.
1	20	5	2	5640	3890	-31.0284	1	3
2	30	5		11655	8562	-26.538	0.5	1
3	40	5		18165	11885	-34.572	0. 5	1
4	50	5		30525	20331	-33.3956	1	3
5	60	5		55005	42475	-22.7797	0.5	3
Average						-29.6627		

Moreover, Table 10 summarizes the difference and improvement obtained due to the changes in release time of jobs. In all the cases compared in Table 10, a significant amount of improvement is observed while the release time generation procedure is changed. The average improvement is reported as around 30%. Although all the four values of α gives better results, the value α = 0.5 helps getting best results in most cases.

Finally, the best values obtained by heuristic combinations 1 and 3 are compared with the results of TSHu and TSD in Table 11. Two set of experiments are performed. The first set has two workers only. In addition, the inputs used in the first set of experiment do not follow the developed function of release time. The first set generates release time from the uniform distribution of [0,100]. On the other hand, the second experiment makes use of the function [0, $(n/m)*P_{avg}*\alpha$] to generate the release times with varying number of workers. It can be noticed that an approximate improvement of 2% and 3% is observed for TSHu and TSD respectively in comparison with the best results of the heuristics.

Table 11: Computational results of all the solution approaches

Case No.	Number of jobs	Number of machines	Number of workers	Best value from heuristics (H)	TSHu	TSD	H vs. TSHu	H vs. TSD
1	20	5	2	5640	5640	5555	0	-1.51
2	20	6		4290	4290	4290	0	0
3	30	5		11655	11450	11250	-1.76	-3.47
4	30	6		8940	8825	8620	-1.29	-3.57
5	40	5		18165	17750	17545	-2.28	-3.41
6	40	6		16320	15895	15580	-2.61	-4.53
7	50	5		30525	29770	29520	-2.47	-3.29
8	50	6		24855	23965	23650	-3.58	-4.85
9	60	5		55005	53765	52405	-2.25	-4.73
10	60	6		43125	41885	41050	-2.88	-4.81
Average							-1.91	-3.42

The experiment (second set of experiment) is then extended up to 20 cases varying the variables (number of jobs, machines and workers) of the model where the release time is generated using the function [0, $(n/m)*P_{avg}*\alpha$]. The value of α is set to be 0.5 as from Table 10 it is apparently observed that this value aids obtaining good results. Each case is duplicated 30 times for the same problem and the average is used for the comparison purpose. The performance of the Tabu search algorithm for dual resources (TSD) is evaluated by comparing the results obtained from combination1 and combination 3 of the heuristics and TSHu. TSD algorithm proved the better performance than those using the heuristics and TSHu almost in all cases although it took longer time than the heuristics and TSHu. The computational performance of the approaches for all 20 cases is presented in Table 12.

Table 12: Computational results of extended experiment

Case No.	Number of jobs	Number of ma-chines	Number of workers	Com 1	Com 3	TSHu	TSD	Com1 Vs TSD (%)	Com 3 Vs TSD (%)	TSHu Vs TSD (%)
1	20	5	2	3712.14	3666.56	3655.45	3575.21	-3.69	-2.49	-2.20
2	30	5	2	8083.57	7974.45	7846.24	7823.75	-3.22	-1.89	-0.29
3	40	5	2	12310.38	12545.32	12243.67	12105.34	-1.67	-3.51	-1.13
4	50	5	2	19389.33	19443.67	19072.64	18921.54	-2.42	-2.69	-0.79
5	60	5	2	25979.02	26562.36	25846.32	25732.79	-0.95	-3.12	-0.43
6	20	5	3	3469.67	3537.33	3410.20	3375.34	-2.72	-4.58	-1.02
7	30	5	3	7433.81	7343.20	7269.15	7181.97	-3.38	-2.20	-1.20
8	40	5	3	11661.34	11524.63	11449.24	11325.33	-2.88	-1.73	-1.08
9	50	5	3	18166.67	17932.23	17844.49	17576.54	-3.25	-1.20	-1.50
10	60	5	3	24374.45	23915.27	23661.56	23193.33	-4.85	-3.02	-1.20
11	20	6	2	3493.27	3425.12	3403.32	3353.12	-4.01	-2.10	-1.48
12	30	6	2	6602.36	6524.95	6445.78	6366.28	-3.58	-2.43	-1.23
13	40	6	2	11587.34	11756.74	11385.14	11201.32	-3.33	-4.73	-1.62
14	50	6	2	17236.59	17464.14	17002.45	16812.78	-2.46	-3.73	-1.12
15	60	6	2	23515.48	23021.75	22732.15	22455.21	-4.51	-2.46	-1.22
16	20	6	3	3233.23	3159.76	3099.10	3054.38	-5.53	-3.33	-1.44
17	30	6	3	6097.46	5996.21	5925.85	5845.45	-4.13	-2.52	-1.36
18	40	6	3	11605.12	11724.56	11432.87	11145.97	-3.95	-4.94	-2.51
19	50	6	3	16353.91	16582.42	16176.38	15828.29	-3.21	-4.55	-2.15
20	60	6	3	22443.23	22132.54	21843.54	21438.46	-4.48	-3.14	-1.86
Average								-3.41	-3.06	-1.38

The table implies the dominance of TSD algorithm over the heuristics and the tabu search heuristic algorithm (TSHu) with an approximate improvement of 3%. Tabu search heuristic algorithm (TSHu) also proves it to be a good tool for scheduling dual resources as it lacks only around 1% from the TSD algorithm. Nevertheless, the algorithm TSD takes very long time in comparison of heuristics and TSHu which is summarized in Table 13. Time requirement of heuristics are only 1.4 seconds whereas around 10 and 50 seconds elapse for TSHu and TSD algorithms.

Table 13: Time requirement of all solution approaches

Case No.	Number of jobs	Number of machines	Number of workers	Time required (sec)			
				Com 1	Com 3	TSHu	TSD
1	20	5	2	1	1	5	30
2	30	5	2	1	1	5	42
3	40	5	2	1	1	8	45
4	50	5	2	2	2	10	53
5	60	5	2	2	2	13	60
6	20	5	3	1	1	5	30
7	30	5	3	1	1	5	42
8	40	5	3	1	1	8	46
9	50	5	3	2	2	10	55
10	60	5	3	2	2	13	62
11	20	6	2	1	1	5	34
12	30	6	2	1	1	6	47
13	40	6	2	1	1	8	52
14	50	6	2	2	2	12	59
15	60	6	2	2	2	16	68
16	20	6	3	1	1	5	35
17	30	6	3	1	1	7	47
18	40	6	3	1	1	12	53
19	50	6	3	2	2	14	61
20	60	6	3	2	2	18	71
Average				4.1	1.4	9.25	49.6

Figure 8 and Figure 9 shows example of the schedules obtained from heuristic and TSD algorithm for 10 jobs on 5 machines with only 2 workers. The TSD algorithm improves the schedule having a flow time of 703 although it finishes all the jobs 5 time units later than the one of heuristic whereas the flow time obtained from the heuristic algorithm is 705. Both the templates reflect the schedule of machines along with the worker. It is observed from the templates that machines remain idle in few slots but workers are mostly busy throughout the scheduling period. So worker is a key factor that has an impact on the scheduling system.

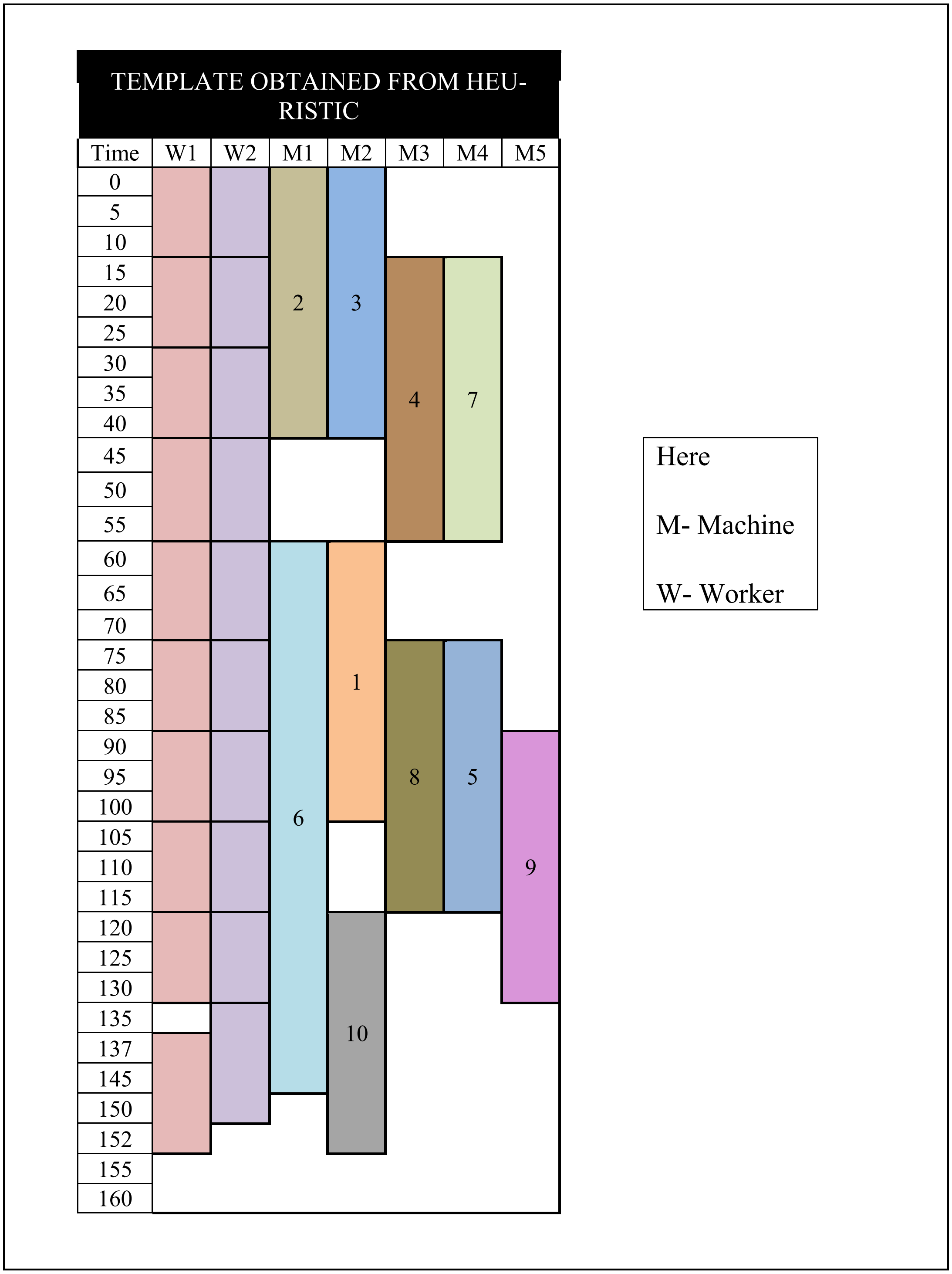

Figure 8: An example of template obtained from heuristic

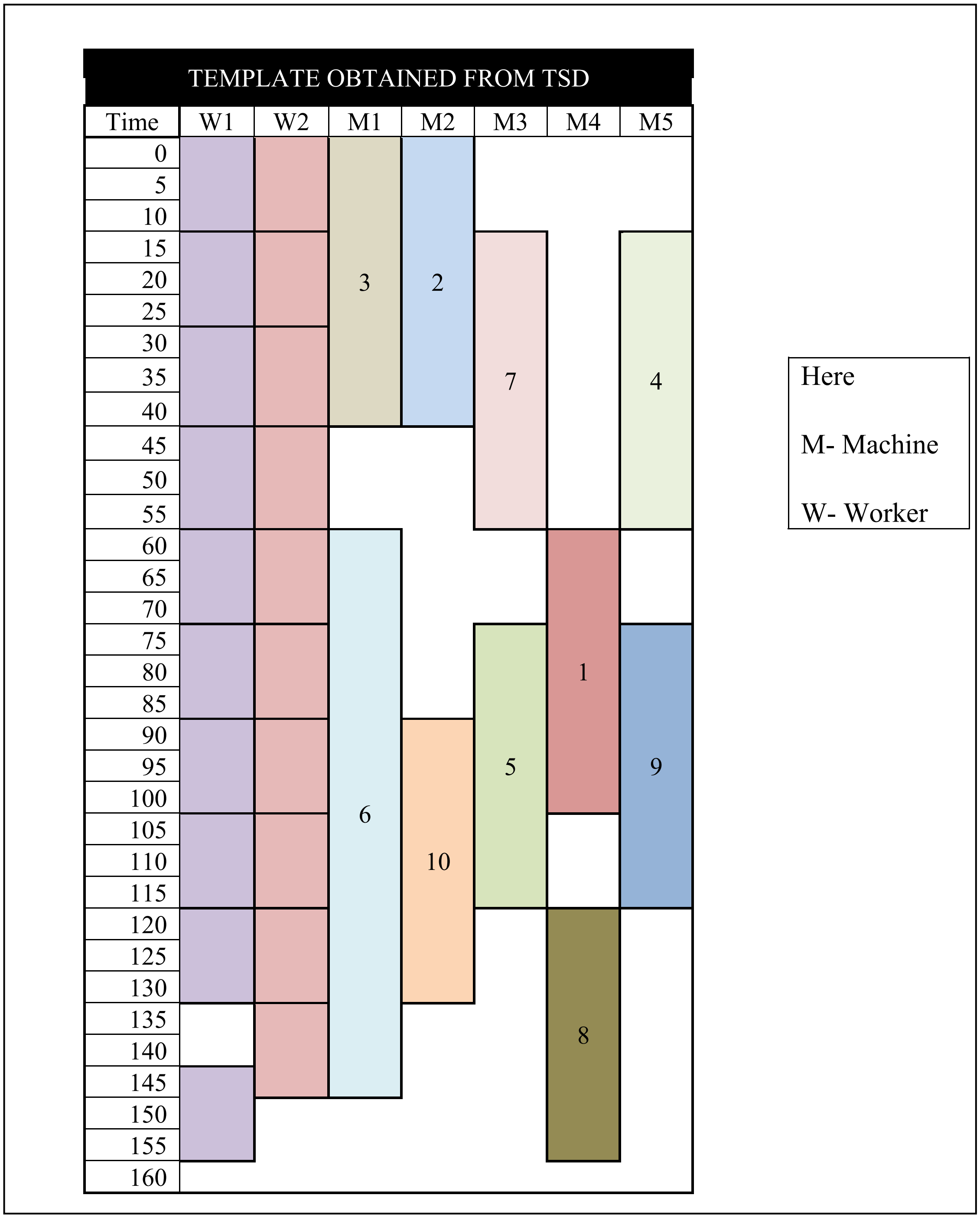

Figure 9: An example of template obtained from TSD

4.5 Case study

A case study is accomplished for the clinic where chemotherapy treatment is provided to cancer patients. The unit consists of 5 stations with 6 chairs in each station. Availability of nurses and the working hours for each station are described below:

i) Station 1: Station 1 has six chairs (numbered 1 to 6) and two nurses. The two nurses work from 8:00 to 16:00.

ii) Station 2: Station 2 has six chairs (7 to 12) and three nurses. Two nurses work from 8:00 to 16:00 and one nurse works from 12:00 to 20:00.

iii) Station 3: Station 4 has six chairs (13 to 18) and two nurses. The two nurses work from 8:00 to 16:00.

iv) Station 4: Station 4 has six chairs (19 to 24) and three nurses. One nurse works from 8:00 to 16:00. Another nurse works from 10:00 to 18:00.

v) Solarium Station: Solarium Station has six chairs (Solarium Stretcher 1, Solarium Stretcher 2, Isolation, Isolation emergency, Fire Place 1, Fire Place 2). There is only one nurse assigned to this station who works from 12:00 to 20:00. The nurses from other stations can help when necessary.

Table 14 reviews and summarizes the nurse allocation and working hours of all stations.

Table 14: Allocation of treatment chairs and nurses' schedule

Station	No of Chairs	Regular Nurses and Working Hour	Float Nurse
Station 1	6	Nurse 1: From 8:00 to 16:00 Nurse 2: From 8:00 to 16:00	Float nurse works from 11:00 to 19:00
Station 2	6	Nurse 1: From 8:00 to 16:00 Nurse 2: From 8:00 to 16:00 Nurse 3: From 12:00 to 20:00	
Station 3	6	Nurse 1: From 8:00 to 16:00 Nurse 2: From 8:00 to 16:00	
Station 4	6	Nurse 1: From 8:00 to 16:00 Nurse 2: From 10:00 to 18:00	
Solarium Station	6	Nurse 1: From 12:00 to 20:00 All the nurses from other station.	

In actual situation it is reported to have on average 135 patients per day in the clinic. The problem is simplified here assuming the clinic has 5 stations with 6 chairs in each station. Each station has its own nurses (2 nurses in each station) without being interchanged within the stations with 125 patients visiting the clinic. Some floating nurses work for the chemotherapy unit in actual case and not all the stations start and finish at the same time. The clinic in real situation has nurses who join the clinic at the middle of the day and work till 8 pm. Thus it is possible to accommodate patients who have treatment times beyond 10 hours. To avoid the complexity of time variation of nurses it is assumed that the stations have 10 hours working time and all the nurses of these stations work from 8 am to 6 pm. To make a balance for that 125 patients are considered in this case study. However, the patients remain common for the entire clinic, i.e. patients are allowed to receive treatment from any of the stations if there is any free chair to accommodate a patient. The nurse of the corresponding station is responsible for the preparations needed for a patient before treatment and also for discharging the patient after the treatment. The clinic works 10 hours each day providing a variety of services that include chemotherapy, blood work, transfusion, hydration, etc. The time required for these treatments varies according to the treatment type. Considering these factors of the clinic, a scheduling template is developed which is presented in Figure 10. Patient arrival and treatment times are obtained from the thesis of a Master's student who worked in the same clinic and reported detailed data of the chemotherapy unit (Ahmed, 2011). These are used as inputs to the TSD algorithm to obtain a scheduling template for the chemotherapy treatment unit.

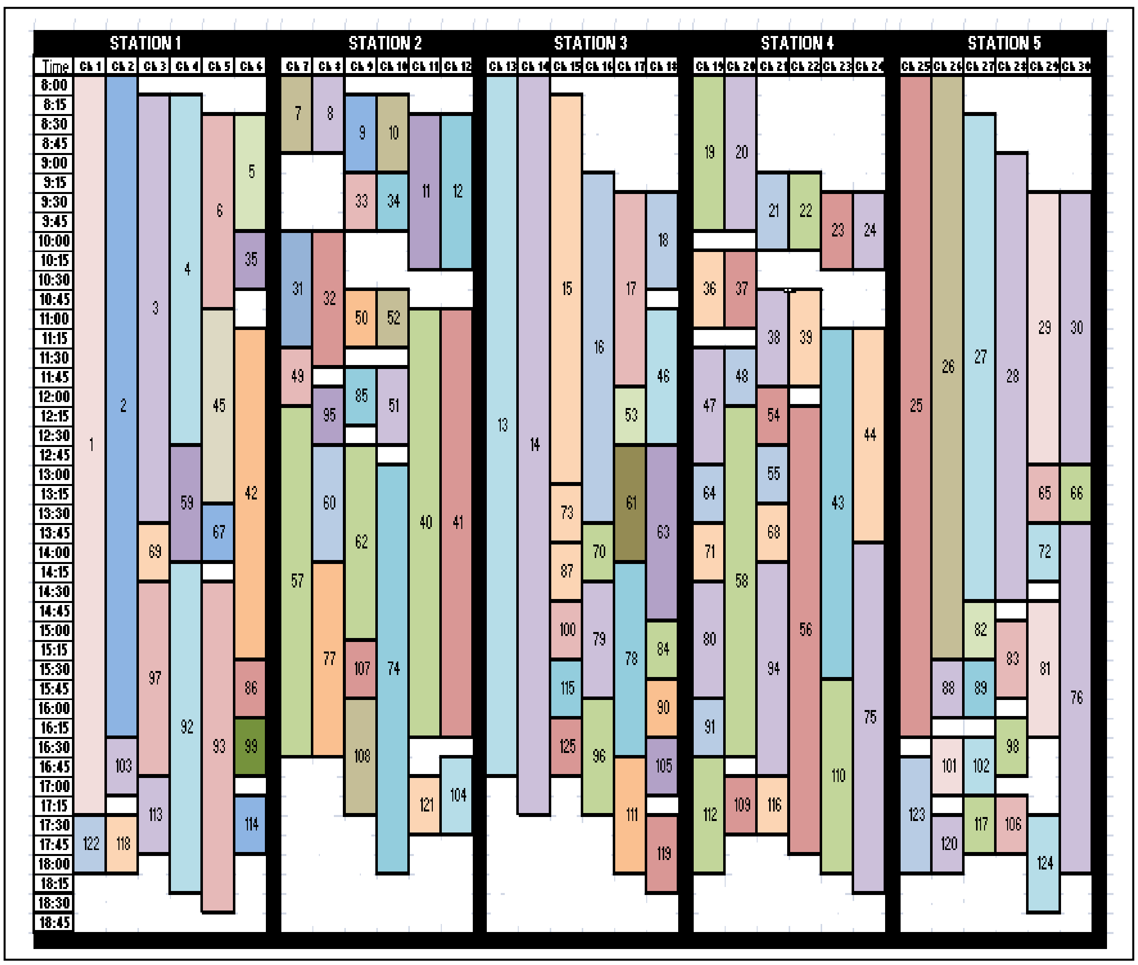

Figure 10: A scheduling template for patients in Chemotherapy treatment clinic

4.6 Conclusion

With the aim of minimizing the total flow time of a chemotherapy treatment unit with dual resources, a metaheuristic: Tabu search is developed in this chapter. The performance of the proposed TSD algorithm is evaluated by comparing its results with heuristic approaches. Moreover, the results obtained from the TSD algorithm are compared with the solutions of a Tabu search heuristic algorithm (TSHu) which is developed for the problem under consideration. In addition, an extensive experiment is executed with the range of release time of patients which is eventually used for the generation of patients' time of arrival for the extended experiment. The experiments of this paper have come to a conclusion that the TSD algorithm outperformed all other approaches considered although it takes longer time than the approaches used to compare with. The algorithm not only suits the healthcare organization but also is applicable in manufacturing sectors where flexible resource (or additional resource) exists to facilitate the processing operations. In future, advanced features like the adaptive memory of Tabu search algorithms can be incorporated to improve the efficiency of the TSD algorithm.

Chapter 5

Conclusion

This study is undertaken to develop an efficient scheduling template of the chemotherapy treatment unit of CancerCare Manitoba, MacCharles site, by minimizing the waiting time of patients. To meet this objective, the scheduling problem is simplified into a single resource problem of identical parallel machines with non common release times. Hrnce, the scheduling problem under consideration has common application in the field of manufacturing and health care. A mathematical model is developed for the scheduling problem of identical parallel machines to identify the optimal solutions for small cases which is followed by a Tabu search algorithm. The Tabu search algorithm has generated results both for the small and large cases as generating optimal solutions for large cases are time consuming. Furthermore, an experiment is executed for jobs release time to make it more realistic. Later on, experiments are conducted with commonly used dispatching rules of scheduling to develop efficient heuristics for identical parallel machines scheduling having dual resources. Further investigation is done to identify the best value of the factor used in the function that generates the release times. Eventually, a heuristic is combined with tabu search algorithm (TSHu) to deal with the dual resource. The experiment is extended by developing the algorithm TSD, which is the Tabu search algorithm dedicated to the dual resources scheduling. This chapter summarizes the research works undertaken in this study and highlights the scope of future research.

5.1 Summary of research

- A chemotherapy treatment unit has two main resources to be considered. The problem of scheduling with two resources is simplified into a single resource scheduling problem in Chapter 3 to avoid the complexities of dual resources. Accordingly, some assumptions are made at the beginning of the chapter. The problem is decomposed as a scheduling problem of identical parallel machines with unequal release time constraint and having single resource. A mathematical model is then developed to represent the scheduling problem which reflects the assumptions. Using the mathematical model, optimal solutions are generated for several small test cases. Afterwards, a Tabu search algorithm is developed for the scheduling problem mentioned above as finding the optimal solution using mathematical model is unattainable for large size problems. Therefore, instead of optimal solutions, near optimal solutions can be obtained using tabu search algorithm. Different treatment durations and arrival pattern of patients are taken into account while developing the Tabu search algorithm (TS). A concept of multiple initial solutions (MIS), unlike the single initial solution commonly used in TS algorithm is introduced in this research. Moreover, the efficiency of the algorithm is improved by introducing further intensification criterion of the algorithm to make it TS+I. The results (of small size problems) obtained from tabu search algorithm are compared with the optimum solutions generated by the mathematical model, to evaluate the performance of the algorithm. The basic TS algorithm has obtained optimal solution for one case whereas TS+I could reache to the optimality in two cases. The performance of the algorithm is also measured by comparing its results with the solutions of modified forward heuristic algorithm (MFHA), found in literature. The tabu search algorithm has led to better solutions than MFHA.

- To make the release time of jobs more realistic, an experiment is further extended in Chapter 3. A function is proposed to generate release times of jobs. The factor α, controls the values to generate release times within the first, second, third and fourth quarter of the total scheduling period. The experiment discloses the fact that, instead of concentrating jobs' arrival at the beginning of the scheduling period, distributing the arrival of jobs throughout the makespan can reduce the total flow time. However, it is concluded that the value of $\alpha = 0.5$ provides best results in most of the test cases.

- In the second portion of the study (Chapter 4), a Tabu search algorithm is developed taking dual resources into consideration to schedule the patients of the chemotherapy treatment unit. Patient scheduling primarily depends on the availability of chairs and nurses. All necessary preparations are perfromed by the nurses while providing treatment to a patient on a chair. Moreover, a patient after completing the treatment procedures needs to leave the system. A nurse has to discharge that patient. Therefore, nurses play an important role to load and unload a patient in the clinic. Three major situations of the clinic are identified and marked as State A, State B and State C in this thesis. These states determine the action of nurses as a nurse has to respond according to the situation of the clinic. The problem again is considered as an identical parallel machines scheduling problem with unequal release time constraint and having dual resources. Heuristics are developed by combining commonly used dispatching rules for the three states. Out of several combinations two heuristics are identified through an experiment that gave better results in comparison to other heuristic combinations. Another experiment is conducted with the arrival pattern of patients which provides a reasonable value of the factor α. By using this value of α, the experiment is further extended to develop algorithms to facilitate the scheduling. The Tabu search algorithm of Chapter 3 is improved to accommodate dual resources by adopting the concept of the heuristic that gave the best results in most of the previous test cases and is named as Tabu search heuristic (TSHu). To assign the second resource (nurse/worker) to patients/jobs, the heuristic approach is used to the solution obtained from the Tabu search (TS) algorithm developed in Chapter 3. Moreover, for improved solutions Tabu search algorithm for dual resources (TSD) is developed by introducing the heuristic at each iteration of the Tabu search algorithm. Twenty test cases are generated to evaluate the performance of the TSD algorithm. The results obtained from heuristics and TSHu are compared with the solutions of TSD. TSD has led to better solutions compared to the other two approaches (heuristics, and TSHu). However, it takes little longer time than TSHu and the heuristics as reported in Chapter 4.

- Furthermore, a template is developed for the chemotherapy treatment unit of CCMB, MacCharles site. The clinic under consideration has five stations with six chairs and two nurses (except station 5) in each station. Additionally, floating nurses help to run the clinic efficiently during the peak hours. The clinic serves for 135 patients every day on average with different working hours of stations. The actual case is simplified here assuming the number of nurses remains two for each station and all the stations follow similar starting and ending time unlike the real situation. Also the patient number is limited to 125 to balance the number of nurses and the working hour. A template for the clinic is then developed using the TSD algorithm.

5.2 Future work

The study of this thesis opens a window towards further research scopes and opportunities.

- The study of Chapter 3 has its bindings and limitations with the constraints selected. Additional constraints for example: due date, set-up time or late arrival of patients can be considered in future to make it similar as the actual situation. Optimal solutions are generated only for 10 small cases in this research. Obtaining the optimal solutions for some of the large cases will be a piece of future work though it is time consuming. Further optimal solutions will provide a scope to evaluate the performance of the Tabu search algorithm generated in this research. Moreover, the Tabu search algorithm developed in Chapter 3 has good scope to improve its efficiency as the results obtained from the algorithm have sufficient differences from the optimal solutions on average. In order to improve the performance of the Tabu search algorithm advanced features like adaptive memory can be applied. However, literature indicates, hybridizing the Tabu search algorithm with other metaheuristics like Genetic Algorithm, Simmulated Annealying or Particle Swarm Optimization has potential to improve the performance of the algorithm.

- In future, all the combinations of heuristics described in Chapter 4 can be explored in search of better solutions. The TSHu and the TSD algorithms can be improved by using the advanced features of the Tabu search algorithm and by hybridizing the Tabu search algorithm which is mentioned above.

- For the case study mentioned in this thesis, the real problem is simplified for the ease of calculations. All the variables and actual situation criterions can be considered for the development of the template as a future work. Differences in starting time of stations and in the nurse availability in each station can have a major impact on the template developed here. Also, the arrival pattern of patients might change due to the changes of nurse's schedule.

References

Ahmed Z. and Elmekkawy T.Y., "Scheduling identical parallel machine with unequal job release time to minimize total flow time", *International Journal of Industrial and Systems Engineering*, *(in press)*.

Ahmed Z., 2011, "Developing an Efficient Scheduling Template of a Chemotherapy Treatment Unit: Simulation and Optimization Approach", Master's thesis, University of Manitoba, Winnipeg, Manitoba, Canada.

Baptiste P., Peter B., Marek C., Christoph D., Svetlana A. K., Francis S., 2007 "The complexity of mean flow time scheduling problems with release times", *Journal of Scheduling*; Vol: 10, pp: 139-146.

Bilge U., Kirac F., Kurtulan M., Pekgun P., 20042004 "A tabu search algorithm for parallel machine total tardiness problem", *Computers & Operations Research*; Vol: 31, pp: 397-414.

Biskup D., Herrmann J., Gupta J. N. D., 2008 "Scheduling identical parallel machines to minimize total tardiness", *International Journal of Production Economics*, Vol: 115, issue: 1, pp: 134– 142.

Canadian Institute for Health Information (CIHI), [http://www.cihi.ca/CIHI-ext-poral/internet/en/Document/spending+and+health+workforce/spending/RELEASE_28OCT10]

CancerCare Manitoba (CCMB), [http://www.cancercare.mb.ca/home]

Cheng T.C.E., Chen Z.L., Shakhlevich N.V., 2002 "Common due date assignment and scheduling with ready times", *Computers & Operations Research*, Vol: 29, issue: 14, pp: 1957–1967.

Chaudhry I. A. and Drake, P. R., 2009 "Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithm", *International Journal of Advanced Manufacturing Technology,* Vol: 42, pp: 581–594.

Chaudhry I. A., Mahmood S. and Ahmad R., 2010 "Minimizing Makespan for Machine Scheduling and Worker Assignment Problem in Identical Parallel Machine Models Using GA" *Proceedings of the World Congress on Engineering 2010*, Vol: III WCE 2010, June 30 - July 2, 2010, London, U.K.

Chu C., 1992 "A branch and bound algorithm to minimize total flow time with unequal release dates", *Naval research logistics*, Vol: 39, issue: 6, pp: 859-875.

Chaudhry I. A. and Mahmood S., 2011, *Electrical Engineering and Applied Computing, Chapter 42.*

Daniels R.L., Hua S.Y., Webste S., 1999 "Heuristics for parallel-machine flexible-resource scheduling problems with unspecified job assignment", *Computers & Operations Research*, Vol: 26, pp: 143-155.

Daniels R. L., Hoopes B.J., Mazzola J.B., 1996 "Scheduling parallel manufacturing cells with resource flexibility", *Management Science*, Vol: 42, pp:1260-1276.

Du J., Leung J.Y-T, Young G. H., 1991 "Scheduling chain-structured tasks to minimize make span and mean flow time", *Information and Computation*, Vol: 92, issue: 2, pp: 219-236.

Franca P. M., Michel G., Gilbert L., Miiller F.M., 1996 "A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times"; *International Journal of Production Economics,* Vol: 43, issues: 2-3, pp: 79-89.

Glover F. and Laguna M. 1997 "Tabu Search".

Grigoriev A., Sviridenko M., and Uetz M., 2005, "Unrelated parallel machine scheduling with resource dependent processing times", *Proceedings of Integer Programming and Combinatorial Optimization. Lecture Notes in Computer Science Vol. 3509*, p 182-195.

Hubscher R, Glover F., 1994 "Applying tabu search with influential diversification to multiprocessor scheduling", *Computers and Operations Research*, Vol: 21, pp: 877-884.

Health Canada, [http://www.hc-sc.gc.ca/ahc-asc/index-eng.php]

Imran Ali Chaudhry, 2010 “Minimizing flow time for the worker assignment problem in identical parallel machine models using GA” *International Journal of Advanced Manufacturing Technology*, Vol: 48, pp: 747–760.

Kashan A. H. and Karimi B., 2009 “A discrete particle swarm optimization algorithm for scheduling parallel machines”, *Computers & Industrial Engineering*, Vol: 56, issue: 1, pp: 216-223.

Kellerer H. and Strusevich V.A., 2003 “Scheduling parallel dedicated machines under a single non-shared resource”, *European Journal of Operational Research,* Vol: 147, pp: 345–364.

Kim C.O. and Shin H.J., 2003 “Scheduling jobs on parallel machines: a restricted tabu search approach”, *International Journal of Advanced Manufacturing Technology*, Vol: 22, pp: 278–287.

Lenstra J.K. and Kan A.H.G., Brucker P., 1977 “Complexity of machine scheduling problems”, *Annals of Discrete Applied Mathematics*; Vol: 1, pp: 343–362.

Lee W. C., Wu C. C., Chen P., 2006 “A simulated annealing approach to makespan minimization on identical parallel machines”, *International Journal of Advanced Manufacturing Technology*, Vol: 31, pp: 328–334.

Li K. and Chu Z. S., 2009 “A backward algorithm for multi-processor scheduling problem with unequal release dates”, Proceedings of the IEEE International Conference on Automation and Logistics, August 5-7, 2009, Shenyang, China.

Li Y., Li H., Yen V. C., 2007 “A new polynomial algorithm for a parallel identical scheduling problem”, *Journal of Control Theory and Applications*, Vol: 5, issue: 4, pp: 345–350.

Li X., Yuerong, C., Yang, S. 2010 “Minimizing job completion time variance for service stability on identical parallel machines”, *Computers and Industrial Engineering*, Vol: 58, issue: 4, pp: 729-738.

Logendran R. and Subur F., 2004 “Unrelated parallel machine scheduling with job splitting”, *IIE Transactions*, Vo:l 36, pp: 359–72.

Manitoba Health Annual Report, [http://www.gov.mb.ca/health/ann/docs/1011.pdf]

Manitoba Budget 2011, [http://www.gov.mb.ca/finance/budget11/papers/summary.pdf]

Min L. and Cheng W., 1999 “A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines”, *Artificial Intelligence in Engineering*, Vol: 13, issue: 4, pp: 399-403.

Mokotoff E., 2004 “An exact algorithm for the industrial parallel machine scheduling problem”, *European Journal of Operational Research,* Vol: 152, pp: 758–769.

Nessah R., Chu C., Yalaoui F., 2007 “An exact method for $Pm/sds, ri/\sum^{n}_{i=1} Ci$”, *Computers & Operations Research*; Vol: 34, pp: 2840 – 2848.

Nessah R. and Chu C., 2010 “Infinite split scheduling: a new lower bound of total weighted completion time on parallel machines with job release dates and unavailability periods”, *Annals of Operations Research*; Vol: 181, pp: 359–375.

Phillips C., Clifford S., Joel W., 1998 “Minimizing average completion time in the presence of release dates”, *Mathematical Programming*, Vol: 82, pp: 199-223.

Po-Chieng Hu, 2004 “Minimising total tardiness for the worker assignment scheduling problem in identical parallel-machine models” *International Journal of Advanced Manufacturing Technology*, Vol:23, pp: 383–388.

Po-Chieng Hu, 2005 “Minimizing total flow time for the worker assignment scheduling problem in the identical parallel-machine models” *International Journal of Advanced Manufacturing Technology,* Vol: 25, pp: 1046–1052.

Po-Chieng Hu, 2006 “Further study of minimizing total flow time for the worker assignment scheduling problem in the identical parallel-machine models” *International Journal of Advanced Manufacturing Technology,* Vol: 29, pp: 753–757.

Po-Chieng Hu, 2006 “Further study of minimizing total tardiness for the worker assignment scheduling problem in the identical parallel-machine models” *International Journal of Advanced Manufacturing Technology,* Vol: 29, pp: 165–169.

Reeves C., 1995 “Heuristics for scheduling a single machine subject to unequal job release times”, *European Journal of Operational Research*, Vol: 80, pp: 397-403.

Ruiz-Torres A. J. and Centeno G., 2007 “Scheduling with flexible resources in parallel workcenters to minimize maximum completion time”, *Computers & Operations Research,* Vol: 34, pp: 48–69.

Ruiz-Torres A. J., Lopez F. J., Ho J. C., 2007 “Scheduling uniform parallel machines subject to a secondary resource to minimize the number of tardy jobs”, *European Journal of Operational Research*, Vol: 179, pp: 302–315.

Sabouni M. T. Y.and Jolai F., Mansouri A., 2010 "Heuristics for minimizing total completion time and maximum lateness on identical parallel machines with setup times", *Journal of Intelligent Manufacturing*, Vol: 21, pp: 439-449.

Sarıcicek and Celik C., 2011 "Two Meta-Heuristics for Parallel Machine Scheduling with Job Splitting to Minimize Total Tardiness", *Applied Mathematical Modelling*; Vol: 35, issue: 8, pp: 4117-4126.

Sang-II K., Choi, H. S., Lee, D. H., 2006 "Tabu search heuristics for parallel machine scheduling with sequence dependent setup and ready times", *Computational science and its applications-ICCSA*, Lecture notes on computer science, 2006, pp: 728-737.

Shim S. O. and Kim Y. D., 2008 "A branch and bound algorithm for an identical parallel machine scheduling problem with a job splitting property", *Computers & Operations Research*, Vol: 35, issue: 3, pp: 863 – 875.

Shim S. O. and Kim Y. D., 2007 "Scheduling on parallel identical machines to minimize total tardiness", *European Journal of Operational Research,* Vol: 177, issue: 1, pp: 135–146.

Skorin-Kapov J., 1990 "Tabu search applied to the quadratic assignment problem" *ORSA Journal on Computing*, Vol: 2, pp: 33-45.

Taillard E., 1991 "Robust taboo search for the quadratic assignment problem", *Parallel Computing*, Vol: 17, pp: 443-455.

Ventura J. A. & Kim D., 2003 "Parallel machine scheduling with earliness–tardiness penalties and additional resource constraints", *Computers & Operations Research,* Vol: 30, pp: 1945–1958.

ViniCius A. and Denise Y., 2000 "Tabu search for scheduling on identical parallel machines to minimize mean tardiness", *Journal of Intelligent Manufacturing*; Vol: 11, pp: 453-460.

Wikipedia, [http://en.wikipedia.org/wiki/Chemotherapy]

Wikipedia, [http://en.wikipedia.org/wiki/Health_care_in_Canada]

Yalaoui F. and Chu C., 2006 "New exact method to solve the Pm/rj/∑Cj schedule problem", *International Journal of Production Economics*, Vol: 100, issue: 1, pp: 168–179.

Appendix A: Identical Parallel Machines Scheduling with Release Time Constraint using Dual Resources

Table 15: List of several combinations of heuristics

	State A		State B		State C	
Combination No.	Primary rule	Secondary rule	Primary rule	Secondary rule	Primary rule	Secondary rule
1	LPT	Random	Random	None	Load if CR<1	Otherwise Discharge
2	LPT	Random	FIFO	Random	Load if CR<1	Otherwise Discharge
3	LPT	Random	Random	None	Discharge first	
4	LPT	Random	FIFO	Random	Discharge first	
5	SPT	Random	Random	None	Load if CR<1	Otherwise Discharge
6	SPT	Random	FIFO	Random	Load if CR<1	Otherwise Discharge
7	SPT	Random	Random	None	Discharge first	
8	SPT	Random	FIFO	Random	Discharge first	
9	ERD	Random	Random	None	Load if CR<1	Otherwise Discharge
10	ERD	Random	FIFO	Random	Load if CR<1	Otherwise Discharge
11	ERD	Random	Random	None	Discharge first	
12	ERD	Random	FIFO	Random	Discharge first	
13	LPT	ERD	Random	None	Load if CR<1	Otherwise Discharge

Table 15 (continued)

14	LPT	ERD	FIFO	Random	Load if CR<1	Otherwise Discharge
15	LPT	ERD	Random	None	Discharge first	
16	LPT	ERD	FIFO	Random	Discharge first	
17	SPT	ERD	Random	None	Load if CR<1	Otherwise Discharge
18	SPT	ERD	FIFO	Random	Load if CR<1	Otherwise Discharge
19	SPT	ERD	Random	None	Discharge first	
20	SPT	ERD	FIFO	Random	Discharge first	
21	ERD	SPT	Random	None	Load if CR<1	Otherwise Discharge
22	ERD	SPT	FIFO	Random	Load if CR<1	Otherwise Discharge
23	ERD	SPT	Random	None	Discharge first	
24	ERD	SPT	FIFO	Random	Discharge first	
25	ERD	LPT	Random	None	Load if CR<1	Otherwise Discharge
26	ERD	LPT	FIFO	Random	Load if CR<1	Otherwise Discharge
27	ERD	LPT	Random	None	Discharge first	
28	ERD	LPT	FIFO	Random	Discharge first	

Table 16: Evaluation of Com1 & Com2 for values of α

Case No.	Number of jobs	Number of machines	Number of workers	Com 1				Com 2			
				α=0.25	α= 0.5	α=0.75	α= 1	α=0.25	α= 0.5	α=0.75	α= 1
1	20	5	2	5284	5105	5177	4232	5284	5105	5204	4232
2	30	5		9613	8562	10576	13221	9613	8562	10576	13221
3	40	5		12458	11885	13124	20702	12458	12524	13124	20702
4	50	5		26845	24347	32697	21872	26845	24347	32697	21857
5	60	5		45672	43861	44634	46354	45672	43861	44634	46354

Table 17: Evaluation of Com3 & Com4 for values of α

Case No.	Number of jobs	Number of machines	Number of workers	Com 3				Com 4			
				α=0.25	α= 0.5	α=0.75	α= 1	α=0.25	α= 0.5	α=0.75	α= 1
1	20	5	2	5351	5064	4928	3890	5351	5064	4928	3890
2	30	5		9725	9582	9988	10973	9725	9582	10576	13221
3	40	5		16546	12458	13288	19847	16546	12458	13124	20702
4	50	5		25637	24278	23757	20331	25637	24278	32697	21857
5	60	5		45691	42475	43624	46855	45691	42475	43624	46855

Table 18: Evaluation of Com5 & Com6 for values of α

Case No.	Number of jobs	Number of machines	Number of workers	Com 5				Com 6			
				α=0.25	α= 0.5	α=0.75	α= 1	α=0.25	α= 0.5	α=0.75	α= 1
1	20	5	2	5352	5295	5204	4232	5395	5063	4864	4232
2	30	5		12354	11297	10576	13221	12647	13246	16667	19436
3	40	5		17823	15497	14536	20702	16964	15257	18349	20632
4	50	5		23675	26523	32697	21857	24825	22642	21419	26475
5	60	5		44294	42475	43582	46345	45264	43126	46920	49256

Table 19: Evaluation of Com7 & Com8 for values of α

Case No.	Number of jobs	Number of machines	Number of workers	Com 7				Com 8			
				α=0.25	α= 0.5	α=0.75	α= 1	α=0.25	α= 0.5	α=0.75	α= 1
1	20	5	2	5395	5063	4864	4232	5134	4537	4465	4232
2	30	5		12647	13246	16667	19436	16542	13295	12465	18364
3	40	5		16964	15257	18349	20632	16288	14694	13536	18525
4	50	5		24825	22642	21419	26475	25321	23145	22687	26388
5	60	5		49264	43126	46920	44256	48354	44765	45215	46555

Table 20: Evaluation of Com9 & Com10 for values of α

Case No.	Number of jobs	Number of machines	Number of workers	Com 9				Com 10			
				α=0.25	α= 0.5	α=0.75	α= 1	α=0.25	α= 0.5	α=0.75	α= 1
1	20	5	2	5134	4537	4465	4232	5450	5295	5150	4365
2	30	5		16542	13295	12465	18364	12463	11357	10645	132462
3	40	5		16288	14694	13536	18525	17283	15564	14628	21781
4	50	5		25321	23145	22687	26388	23285	26647	32666	21857
5	60	5		48354	44765	45215	46555	44294	43675	45952	46320