Optimization of the Distributed Permutation Flowshop Scheduling Problem

by

Arshad Ali

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 Engineering

University of Manitoba

Winnipeg

Abstract

There are $(n!)^m$ possible solutions for scheduling jobs in a flowshop. Keeping the same schedule for all machines of a factory, brings possible solutions to $n!$. That's the reason of knowing this problem as *permutation flowshop* scheduling problem. In case of distributed permutation flowshop, F possible permutation flowshops are considered for scheduling simultaneously, making it a *distributed* permutation flowshop scheduling problem (DPFSP).

Distributed permutation flowshop scheduling problem consists of solving two problems simultaneously, allocation and sequencing of jobs for each flowshop. In this thesis, distributed permutation flowshop scheduling problem is studied for total flow time and makespan objectives. Additional constraints of no-wait and heterogenous nature of the factories are also considered while solving DPFSP for more realistic problems. The problems are solved by using mathematical model, construction heuristic and tabu search (TS) metaheuristic. Addition of insertion cost matrix and improvement scheme helped achieve improved results for the problems. Extensive numerical experiments are conducted to illustrate the efficiency and validity of proposed algorithms.

The solutions of problems are useful to the decentralized, geographically scattered plants. It may help reduce manufacturing cost, organizational risk and can help improve quality of products. Current research improved results by 0.167% for homogeneous DPFSP problem with total flow time objective. Similarly, improved results by 4.77% for no-wait heterogenous DPFSP problem with makespan objective.

Keywords*: parallel flowshop, distributed permutation flowshop, tabu search, total completion time, makespan, no-wait heterogenous DPFSP*

Acknowledgements

In the name of Allah the most beneficent, the most merciful. Neither a leaf can fall without His order, nor a sand particle can fly. I thank Him for limitless bounties He bestowed upon me.

Some projects, like my thesis cannot be completed without help and support of people around. I am thankful to all of those who helped me to complete this task. This thesis was not possible in its current form without constant technical and moral support of my advisor Dr. Tarek ElMekkawy and co-advisor Dr. Yuvraj Gajpal. I am thankful for their profound trust, encouragement, guidance and patience. Professional skills and technical knowledge I have learned during this work will help me throughout my career in future.

I would like to record my sincere thanks to my committee members Dr. Qingjin Peng and Dr. Srimantoorao S. Appadoo (Department of Supply Chain Management) for their valuable suggestions, feedback and their availability for any official requirement. I thank my colleagues and my friends for their support and motivation. I thank Bernice Ezirim (Department of Mechanical Engineering) and Pam Yarushinsky (Faculty of Graduate Studies).

I express my sincere and deepest gratitude to my father (Shakir Ali), brothers and sisters. I am very grateful to my wife, my children Fatima, Abdullah and Zahra. This achievement of mine was not possible without help and support of you all.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

In today's globalized environment, a competitive strategy is required for any business to survive regardless of nature of the business. Same is the case for manufacturing entrepreneurs which are integrated conventionally. These integrated companies have central production planning and control mechanism dealing with vertically integrated set ups. However, current market requires a system which can co-op with market demands such as customized products, product variation, lower cost, quick response and shorter lead times. It requires development of a comprehensive strategy which can deal with all these variables for a business.

The current work focuses on one of these strategies known as distributed manufacturing. This distributed manufacturing can be either under the same roof or at distant location. In case of distant locations, it helps reduce the work related to material supply and take benefit from information sharing. Distributed manufacturing also helps to reduce manufacturing cost, associated risks and improve product quality (Chan and Chung, 2005).

1.1. Recent issues and motivation

World market economy is highly competitive than ever before. In this environment open market access, razor-thin price margins, higher commodity costs, smaller life cycle, demand for product versatility and product customization are some of the challenges faced by conventional manufacturing. Central planning and manufacturing system are no longer capable to deal with such kind of problems. Not being robust, conventional manufacturing is incapable to comply with this challenging market due to different reasons. These systems are bound to follow the tiered, rigid management styles leading to longer path for any changes to accommodate. These complexities are one of the reasons of not using "customer first" approach to strive for higher market share. To deal with these issues and to accommodate flexibility and dynamic nature of the upcoming market demands, introduction of innovative manufacturing systems is required. A short description of some of these modern manufacturing systems is given here. Our current research is also a contribution to the same group of manufacturing systems.

A *flexible manufacturing system* (FMS) is defined as "a computer-controlled configuration of semi-independent workstations and a material handling system designed to efficiently manufacture more than one part number" (Charles Stark Draper, 1984). Dynamic nature of the system in FMS helps to process different types of jobs simultaneously. In 1992, Ueda Kanji discussed about a new concept of decentralized manufacturing based on biological phenomenon of metaphors known as *bionic manufacturing system* (BMS). The BMS uses parallels of biological mechanism for advanced manufacturing techniques. This system allows factories to reduce their supply chain related expenses and allows to be responsive quickly to any required changes. By this systems organizations can reduce associated risks, taxation, transportation and production cost and can improve product quality (Kahn, Castellion, and Griffin, 2004).

Fractal factories is another concept of modern manufacturing (Warnecke, H. J. 1993). This technique can adapt itself against external affects due to dynamic nature. *Agile manufacturing* refers to a system which can quickly respond to any customer demand or market change and rapidly reengineer the process to customize the product or process as required (Goldman et al., 1994). *Holonic manufacturing* is one of advanced manufacturing systems introduced by Koestler in 1968. Holonic manufacturing system is known as decentralized and dynamic system.

Reconfigurable Manufacturing System (RMS) (Koren et al., 1999) is a mechanism to deal with market changes and is cost effective. Adapting new technology, continuous improvement to accommodate innovations and market demand help RMS improve continuously.

Distributed manufacturing is a multi-factory system developed to be competitive and to deal with problems related to single factory manufacturing (Peklenik, 1992). An extension of the same for permutation flowshop is considered by Naderi and Ruiz (2010). Current research extends the same base further to deal with different constraints and objective functions.

1.2. Problem statement

Distributed permutation flow shop is an extension of permutation flowshop with an added constraint of jobs assignment. These two problems are required to be solved while dealing DPFSP problems. In this thesis, algorithms are developed to solve these problems in homogenous and heterogenous environments. In homogenous environment, all parameters are deterministic. Factories are considered similar and the output will be same regardless choice of factory, a job is assigned. This problem is solved for total flow time objective function, which obviously lead to reduction in processing time of a job.

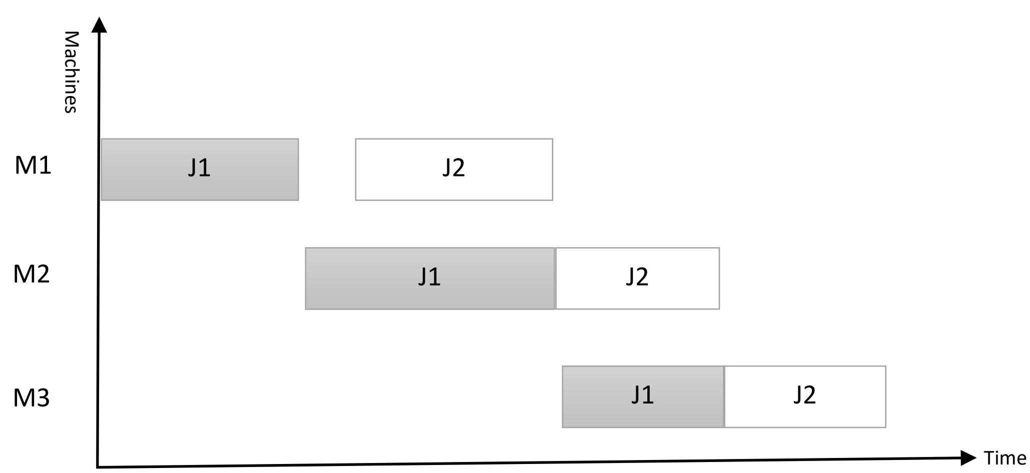

Figure 1 Example of no-wait flowshop

In another problem of heterogenous DPFSP, each machine of every factory has a different machine speed with the added constraint of no-wait. This heterogenous problem is solved for makespan objective function. Consideration of wait is useful in processing of different products including steel, plastic, glass etc. Figure 1 explains about no-wait flow shop scheduling.

1.3. Research objective

From above discussion it can be realized that further research on DPFSP problem is required. The DPFSP has wide practical application but available literature indicates that work in this field is limited till today as compared to other scheduling related problems. This scheduling problem is an NP-hard in nature which is not possible to solve by using conventional mathematical programing solution methods. Consequently, development of algorithms to generate good results within reasonable time is essential.

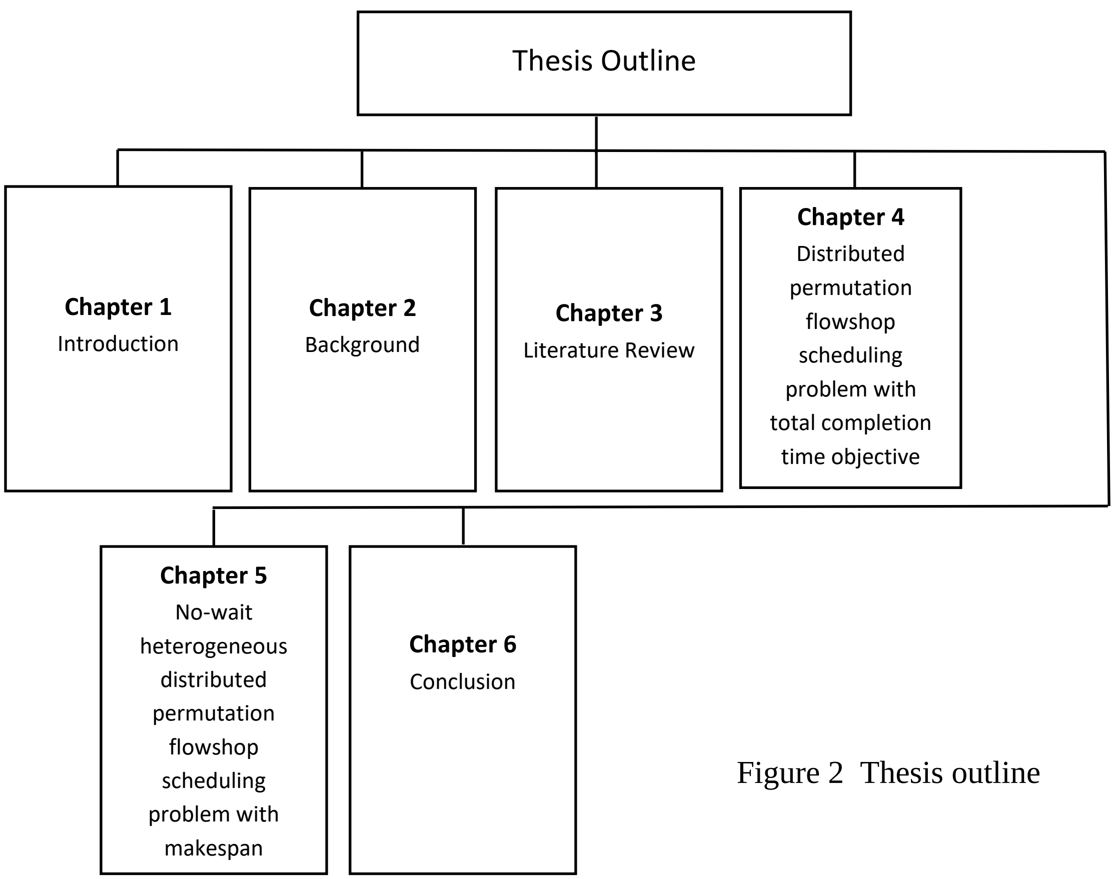

Figure 2 Thesis outline

This research discusses various DPFSP scheduling problem from analytical perspective. Analytical discussion lead to discuss further applicable constraints and assumptions. These constraints and assumptions lead to development of metaheuristic algorithms which are developed to deal with the problems under consideration. The results achieved using these algorithms are compared efficiently and effectively using problems from the literature. The proposed method algorithms proved to be competitive to those of available in literature.

1.4. Thesis overview

This thesis is organized as follows; in chapter 1, we have already presented a brief review of scheduling terminology and scheduling classification. Different solution methods for scheduling problems are also explained here. Chapter 2 presents an explanation of some manufacturing environments and distributed permutation flowshop problem. Different solution techniques are also outlined here. A literature review about the problem and related topics is presented in chapter 3. Chapter 4 is based on analysis of distributed permutation flowshop scheduling problem with total completion time objective. In this chapter, a set of homogenous factories is considered to schedule jobs for total flow time objective. A mathematical model is also presented for the problem. A metaheuristic, tabu search is used to solve the problem.

Chapter 5 presents an extension of the same DPFSP problem with no-wait constraint for makespan objective. Here, problem description is presented detailing about the nature of factories set up where each machine may have different speeds, leading to more realistic problem approach. In this scenario of heterogenous factories, processing time of each job depends on the factory where it will be processed. A conclusion of current research is presented in Chapter 6. This includes description of search findings, results and future work.

CHAPTER 2

Background

Centralized scheduling and planning concepts are not valid in today's market, requiring more customized products/services. The demand of current dynamic market can only be fulfilled by a system which is more flexible and responsive to accommodate the changing requirements. Different nodes of manufacturing/service facilities are required to be developed by either of merger, launching new locates or acquisitions. In this new decentralized environment where facilities can not necessarily be in the same whereabouts, companies work independently and have different working models using which they can either access details/information of other sites/locations or not. Each manufacturing site is responsible to perform some of the tasks independently not only enjoying unique advantages in term of either labour, technology etc. but also facing unique limitations for that site.

This chapter presents a basic review of scheduling, types of scheduling, manufacturing systems and solution methods for scheduling problems. These topics are covered in a comprehensive manner just to introduce the reader with the topics which clarify the following literature and research.

2.1 Operations scheduling

Scheduling deals with the allocation of resources, typically machines, to tasks (commonly referred to as jobs) over time with the goal of optimizing a given objective (*Pinedo,* 2012). Scheduling objective can be either minimization (makespan, tardiness, flowtime) or maximizing (profit margins) of subjected function. Almost every area of our daily life ranging from office staff, transportation, aviation, production, communication to health service is subjected to scheduling somehow. Johnson's (1954) work was pioneer publication for scheduling followed by thousands of publications on the same. Basic information related to scheduling are job's processing time, release time/date and due date/time.

A basic scheduling problem is identified by three notations known as triplet which are α | β | γ. A scheduling problem is detailed according to the information provided by triplet. Machine environment is described by α field, β field informs about processing characteristics and γ contains the information about the objective function of considered scheduling problem.

Table 1 Examples of machine environment (α), constraints (β) and objective functions γ

Machine environment (α)		Constraints (β)		Objective functions (γ)	
1	Single machine	*rj*	Release dates	*C*max	Makespan
Pm	Identical machines in parallel	*Prmp*	Pre-emptions	*L*max	Maximum Lateness
Qm	Machines in parallel with different speeds	*prec*	Precedence constraints	$\sum wjCj$	Total weighted completion time
Rm	Unrelated machines in parallel	*Sjk*	Sequence dependent setup times	$\sum wj(1-e^{-rCj})$	Discounted total weighted completion time
Fm	Flow shop	*Fmls*	Job families	$\sum wjTj$	Total weighted tardiness
FFc	Flexible flow shop	*batch*(*b*)	Batch processing	$\sum wjUj$	Weighted number of tardy jobs
Jm	Job shop	*Brkdwn*	Breakdowns		
		Mj	Machine eligibility restrictions		
		Prmu	Permutation		
		block	Blocking		
		Nwt	No-wait		
		Rcrc	Recirculation		

2.2 Scheduling problem classification

According to machine environment, a scheduling problem can be classified in three main classes, which are;

A. Single machine

B. Multiple machines in single stage

C. Multistage multi machine

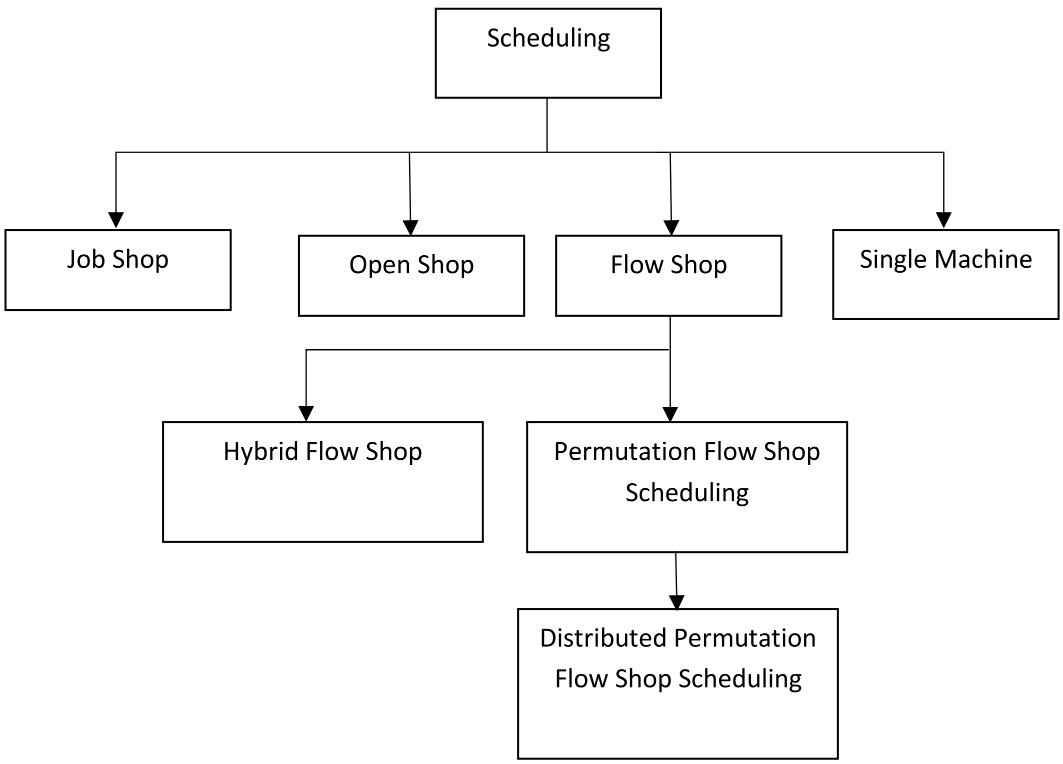

Figure 3 Hierarchy of distributed permutation flowshop scheduling problem

2.2.1 Single machine scheduling

In this type of scheduling environment, a single resource or machine deals with all tasks to be processed. Scheduling theories and rules applied to single machine scheduling problem play important role for understanding of other related scheduling problems as it's the simplest of all environments. Following assumptions are considered while scheduling for a single machine problem (Hatami, 2016);

- Machine is continuously available during scheduling period
- A machine can process only one job at a time
- Processing times of jobs on machine are deterministic positive integer
- Jobs can be planned in either of pre-emptive or none pre-emptive case as per requirement

2.2.2 Parallel machine scheduling

Parallel machine scheduling problems can be broadly divided in two major categories of 1) parallel machines at single stage, and 2) parallel machines at multiple stages.

Parallel machines at single stage

Theoretically it's a special case of flexible flow shop and it is a generalization of single machine problem. This type of problems contains single production or service unit having multiple parallel machines. Scheduling on parallel machine can be considered as two stage planning. In first stage, allocation of jobs is done to all machines. In second stage of scheduling, sequencing of jobs allocated to each machine is performed. It is due to this reason that theoretically parallel machine scheduling is also known as special case of flexible flow shop.

Parallel machine scheduling is applicable in case if *n* single-operation jobs are available simultaneously at time zero to be processed on either of *m* available machines. A job can be processed at most on one machine at a certain given time. It doesn't allow to process more than one jobs on a single machine simultaneously.

Due to its wide practical application in real world, parallel machine scheduling has been widely studied by researchers (Lin et.al. 2011). Parallel machines disposed in this kind of scheduling problem can be either identical (P), unrelated (R) or uniform related (Q). In parallel identical machines, processing time of a job is same on all machines and can be scheduled on multiple parallel machines. Unrelated machines scheduling environment contains unrelated machines with different characteristics resulting in different processing time p_{ij}. Machine perform with different speeds in case of parallel uniform related machines. The processing time of a job depends on machine speed and can be calculated using $p_{ij} = p_j/s_i$ relationship, where s_i indicates speed of machine i.

Parallel machines at multiple stages

It is an extension of multiple machines in single stage with an addition of multiple stages or stations for processing of same set of jobs in a flow. This environment can be further divided into flow shop, flexible flow shop, job shop and open shop scheduling problems.

2.2.3 Open shop

Open shop is an environment where each job i is processed in a way that it goes through m set of machines to complete m operations. Every job is independent of every other job which leads to no-precedence relation between operations. In open shop, jobs can be processed in any order. Jobs are not required to be in need of any order as its irrelevant in this environment. A machine can process only one job at a time and each job can be processed at most by a single machine at any given time.

2.2.4 Job shop

In job shop, jobs are processed through a series of machines just as flow shop with the distinction that each job doesn’t need to follow the same route as predecessor or successor. Each of i job from a set of n (1… n) jobs is required to pass through a distinctive set of m machines ($M_1, \ldots M_m$) in order to be finished. This route can be distinctive for every job. A feasible schedule is required to be identified in order to optimize the desired objective function. In job shop environment, comparatively more constraints are involved as compared to open shop such as; each job can be processed only at one machine at certain time, similarly, a machine can process only one job at a time.

2.2.5 Flow shop scheduling problem

In flowshop scheduling problem, m operations are performed in series on each job i with processing times p_{ij} where j = 1… m. All jobs follow the same sequence and are required to be processed on each machine in the unidirectional flow. Each machine is designed for specific operation. All jobs are available at time zero where each machine can process at most one job at a certain time and vice versa. With all above conditions, the problem is to find a job sequence π_j for each machine j (Gajpal & Rajendran, 2006).

Flow shop problems are also generalised as permutation flowshop in literature. It's due to the fact that a random number of sequences are possible on each machine in the flow which can be as many as $(n!)^m$. It is difficult to find an optimum or near optimum sequence from these enormous number of solutions. Research focused on reducing these possible solutions lead to assumption of having one schedule for all machines in the sequence as that of first machine. It means that all machines in the sequence will have the same sequence of jobs as that of first machine. This assumption reduced the number of possible sequences to be $n!$. It is due to this reduction of schedules form $(n!)^m$ to $n!$ that flow shop is also known as permutation flowshop.

2.2.6 Mixed shop scheduling problem

Mixed shop problem is a combination of open and job shop problem where both open shop and job shop jobs are available. Distributed permutation scheduling problems deals with different type of manufacturing as those of above-mentioned problems. Current research deals with scheduling of this type of distributed manufacturing.

2.2.7 Distributed scheduling problems

Multiple factors, including benefits of computer networks, impact of globalization and demand for showing presence of companies in different regional markets has driven both suppliers and manufacturers to develop their distributed manufacturing set-ups scattered geographically. This kind of environment requires more responsiveness to emergencies, flexibility and ability to satisfy the unexpected dynamic market needs. Since conventional centralized manufacturing and resource planning is unable to deal with these types of issues. Distributed manufacturing also known as multi-site manufacturing is one of the solutions to deal with these problems. Distributed manufacturing is more flexible and has the capacity to be more responsive.

These distributed manufacturing set-ups (mostly distributed geographically) are responsible for performing operations, unique for that site. This unique operation are based with consideration of labor cost, available technology, raw material availability, taxation, government policy, availability of transport facilities etc. (Chan and Chung, 2013). Complexity of distributed scheduling problem is obviously higher than the single centered scheduling problem. In single centered manufacturing, scheduling of jobs includes only finding a sequence of jobs allocated to the location. Whereas in case of distributed manufacturing, the scheduling problem includes finding solution of two problems; 1) allocation of jobs to each facility and 2) sequencing the jobs for each facility. Readers can be referred to Kuhnle (2009) for further studies about the topic.

2.3 Solution approaches for scheduling problems

There are various methods of solving these scheduling problems. Some of these methods are discussed here in this section. These are 1) Heuristic Methods 2) Metaheuristics Methods 3) Exact Solution Methods

A heuristic method can be defined as a technique which is likely to discover a solution but necessarily not an optimal one at a reasonable computational cost. Heuristic methods are based on simple thoughts targeted to search solutions easily. These heuristics are also referred to as scheduling or dispatch rules (Pinedo, 2012). Usually, heuristic solutions are problem-specific which indicate that a heuristic method which is suitable for one problem cannot be used to solve a different problem (Reeves, 1993). A heuristic method usually operates in iterative manner where every new iteration operates to find a new solution within search space. This new solution might be better from the previously available best solution until now. After completion of iterative procedure, heuristic method provides the best available solution.

As per description of heuristic method mentioned above, every time a problem appears, a heuristic method customized to solve the problem needs to be developed. Solution method to such problem has revolutionized with advent of metaheuristic, which is a strong tool to solve operations research problem. Metaheuristics are algorithms designed to solve higher level and wide range of scheduling problems. It is an algorithm designed to solve a wide range of hard optimization problems. The family of metaheuristics includes but is not limited to tabu search, simulated annealing, genetic algorithm, ant colony optimization particle swarm optimization, iterated greedy. Most of the metaheuristics are inspired from nature or physical processes. For example, particle swarm optimization is inspired from flocking behaviour of birds (Sahu, Gajpal, & Debbarma (2018)) and ant colony optimization is inspired from foraging behaviour of ants (Li, Gajpal, &

Bector (2018)). Also, it can be noticed from Table 2 that metaheuristics got more interest in last three decades. The development of metaheuristics is fostered with the increase of processing power and hardware of computers.

Table 2 Examples of machine environment (α), constraints (β) and objective functions γ

Metaheuristic	Proposed in	Proposed By	Inspired by
Ant colony optimization	1992	Dorigo	Behavior of real ants
Artificial bee colony algorithm	2005	Dervis Karaboga	Behavior of honey bees
Genetic algorithm	1988	Koza	Chromosome behaviors
Particle swarm optimization	1995	Kennedy and Eberhart	Swarm of Insects Flocks of Birds School of Fish
Simulated annealing	1983	Kirkpatrick et al	Annealing in metallurgy
Tabu search	1986	Glover	Forbidden-Scared

Exact solution methods are the only solution methods which promise to provide an optimal solution. Mathematical programing (MP) and branch and bound (B&B) are well known types of exact solution methods among other available. The MP is used for wide range of disciplines including scheduling. Two well-known examples of MP in scheduling are linear programming (LP) and mixed integer linear programing (MILP). The LP is known by this name as all of constraints and objective functions are linear in nature. However, in some situations it becomes necessary to restrict the decision variables to binary or integer. If all variables need to be integer, it is called a (pure) integer linear program. If all variables need to be 0 or 1, it is called a 0-1 linear program. The MILP are often much harder to solve than LP. The MILP benefits in a way that it provides exact solution instead of approximate.

Branch and bound (B&B) is based on 'divide and conquer' strategy. Feasible regions of the solutions are further divided, if required. Further division results in lower and upper bounds. The B&B relies on two routines that compute lower and upper bound. Upper bound is found by a local optimization method.

CHAPTER 3

Literature Review

The permutation flowshop scheduling problem received attention after the founding research work of Johnson (1954), Osman and Potts (1989), and Shabtay et al, (2013). In a conventional shop environment, all jobs are processed through the series of machines in a single factory (Gajpal, Rajendran, & Ziegler)2006)). In contrast, a distributed production model involves the processing of jobs through one of the different available factories (Fan, 2010). The literature on distributed manufacturing is scarce compared to classical shop scheduling problems. A distributed job shop scheduling problem has been analyzed by different authors including Jia et al. (2002, 2003), who used a basic genetic algorithm; Chan et al. (2005) using an adaptive genetic algorithm, and De Giovanni (2010), using an improved genetic algorithm. Similarly, Naderi and Ruiz (2010) are the first one to propose distributed permutation flowshop scheduling.

Distributed permutation flow shop environment can be either homogenous or heterogeneous. In homogeneous environment, machines are identical factory to factory with no difference of processing time of a job in either of the factory. It is due to the symmetry of machine speed, technology and nature of machines. Value of objective function will be same regardless of selected factory. Whereas in case of heterogeneous arrangement, machines are unidentical in terms of either speed, technology, setup time etc. The processing time of jobs and consequently, value of objective function varies factory to factory. Literature for both homogeneous and heterogenous DPFSP problems is presented here.

3.1 Homogenous Factories

Naderi and Ruiz (2010) analyzed DPFSP for minimizing the makespan objective function. They developed six alternative mixed integer linear programming (MILP) models for the problem. Two iterative methods based on variable neighborhood descent (VND) were also presented. Naderi and Ruiz (2010) introduced 420 small size problem instances and 720 large size problem instances. These instances were later used by other researchers to evaluate the performance of their algorithms.

The DPFSP problem has been solved for makespan minimization using different algorithms such as the discrete, electromagnetism mechanism algorithm (Liu and Gao, 2010), genetic algorithm (Gao and Chen, 2011), variable neighborhood descent (VND) based algorithm (Gao et al, 2012), tabu search (Gao et al, 2013), hybrid immune algorithm (Xu, 2014) and an estimation of distribution algorithm (Wang et al, 2013). Fernandez-Viagas and Framinan (2015) proposed a bounded-search iterated greedy (BSIG) algorithm for the same, and the results were compared with an estimation of distribution algorithm (EDA), a tabu search (TS) and a modified iterated greedy (MIG) algorithm. Naderi and Ruiz (2014) analyzed the problem by applying an evolutionary algorithm known as scatter search, compared the effectiveness and efficiency of the existing methods published from 2010 to 2014, and presented improved results.

After the work of Naderi and Ruiz (2014) more papers dealing with solving DPFSP for a makespan objective, were published. Komaki et al. (2015) applied a general variable neighborhood search (GVNS). Li and Chen (2015) used a genetic algorithm, and Wang et al. (2016) applied a hybrid discrete cuckoo search (HDCS) algorithm for solving the problem. A novel, chemical reaction optimization was applied by Bargaoui and Driss (2017) who presented some of the improved results for the DPFSP problem. Finally, Ruiz et al. (2019) used the iterated greedy method to provide the state-of-the-art results for a makespan objective.

Fernandez et al. (2018) were the first to analyze the DPFSP problem for total completion time objective by using an evolutionary algorithm (EA). A comparative analysis of the proposed algorithm was performed with three most efficient algorithms used to solve the DPFSP for makespan objective. They modified the bounded search iterated greedy (BSIG) algorithm, iterated greedy (MIG) algorithm, scatter search (SS) and evolutionary algorithm (EA) from the makespan objective to the total completion time objective. The results indicate that the performance of EA is better than those of other algorithms.

Wang et al. (2016) studied the DPFSP under machine breakdown and used fuzzy logic-based hybrid estimation of distribution algorithm to solve the problem for makespan criteria. To reduce the makespan, Duan et al. (2016) analyzed the DPFSP with flowline eligibility constraint where at any given time every factory is not available. A hybrid estimation of distribution algorithm (EDA) with addition of a heuristic and a local search was proposed. Li et al. (2016) incorporated a transportation constraint with DPFSP, which according to them is pioneer work in subjected area. They considered different vehicle loading capacities and timetable schedules for different factories and solved the problem for makespan objective. Deng et al. (2016) introduced a variation of DPFSP with makespan and total carbon emission criteria. A competitive memetic algorithm (CMA) was used to analyze the effect of parameters on objective function.

3.2 Multi-objective optimization

Though most of the DPFSP related research is focused for single objective function but there is some published work which is focused on multi-objective optimization of DPFSP. Cai et al. (2017) studied DPFSP with additional constraints of transportation and eligibility. The problem was analyzed for three objective functions including makespan, total cost and maximum lateness. Initially, for makespan and maximum lateness, eight heuristics are proposed while for total cost a greedy algorithm has been proposed. Finally, an improved non-dominated sorting genetic algorithm is proposed to solve multi objective DPFSP problem.

Deng & Wang (2017) presented work with focus of makespan and total tardiness criteria. They used competitive memetic (CMA) algorithm to solve the problem. Two separate populations are initialized dealing with two objectives. A Pareto based approach is used to deal with conflicting objectives. Three performance metrics are used to evaluate the algorithm including, overall non-dominated vector generation, the number of distinct non-dominated solutions and *C* metric (to compare two non-dominated solution sets). The *TS* is used to measure how evenly the solutions are distributed. Results are compared with rand and no-dominated sorting genetic (NSGA) II algorithms.

Wang et.al (2018) studied DPFSP for multiple objectives of makespan and energy efficiency. They considered ecological perspective of the problem and solved the problem using a knowledge based cooperative algorithm (KCA). The results are compared with competitive memetic algorithm and non-dominated sorting-based MOEA, known as NSGA-II algorithm. The KCA is indicated as better performer compared to other algorithms in terms of solution quality and diversity.

3.3 DPFP with no-wait constraint

An enormous amount of literature is available for permutation flowshop scheduling problem (PFSP) but literature with consideration of no-wait is scant. Areas of no-wait manufacturing includes food, medicine, plastic, metal, chemical etc. They are processed in an environment where unfinished jobs cannot be delayed between start to end of processing. In simple words, after starting processing of a job on first machine, it cannot be stopped on consecutive machines unless all operations are finished. Such kind of requirement arises in situations where formation of material is done while it is hot. Cooling, if delayed, makes it difficult to form. A well-known example is processing of metal where molten metal needs to be finished before it gets cooled. This system of not allowing to delay jobs is known as no-wait. In a no-wait conventional flowshop environment each job is to be processed without interruption between machines. Once a job is started, a job must be continuously processed without interruptions and preemption. To follow this no-wait constraint, a job is delayed on first machine before processing starts. Extension of DPFSP with no-wait is known as distributed no-wait flowshop scheduling problem (DNFSP).

In petroleum refineries, rolling and chemical activities are required to be completed with no-delay to avoid related penalties (Arabameri and Salmasi, 2013). Different operations of steel manufacturing including ingots, unmolding, reheating, soaking and preliminary rolling are supposed to follow each other without any delay. Hot-work-progress during this manufacturing process should not wait between either of consecutive processes (Hall & Sriskandarajah, 1996). For example, a slab of steel is heated to certain temperature before it starts processing. The temperature will reduce considerably if it waits in front of any machine during processing. Re-heating it, in worst case scenario, will cause a big energy loss and delay in achieving the objective. Consequently, it should be considered to schedule in a no-wait environment.

Application of no-wait scheduling in surgery was analyzed by Wang, Han, Zhang, and Zhang (2015). They considered it as no-wait flowshop having three machines with application of predictive –reactive scheduling. The objective was to accommodate the influences of surgery’s arrival as per plan to optimize the efficiency. Similarly, no-wait flexible flowshop concept is used by Wang, Tang, Pan, and Yan (2015) to plan a surgery for a scarce medical resources to reduce the operating cost. An algorithm with insights into cost and planning the process in two phases was developed by the authors. First phase of research was focused to find out if the operation can be planned within the defined planning period. The second stage consist of scheduling the daily surgery schedule. Patient scheduling is also mentioned as no-wait flowshop scheduling problem by Hsu, De Matta, and Lee (2003).

Flight scheduling to minimize the maximum number of aircrafts within single air traffic section at any time is also considered to be no-wait scheduling problem (Kim, Kröller, Mitchell, and Sabhnani, 2009). Similarly, aircraft sequencing problem for landing with the objective of maximizing the number of landing aircraft in the time unit is also considered to be no-wait scheduling problem (Bagassi, Francia, and Persiani, 2010). Scheduling of metro trains is another application of no-wait scheduling problem. Mannino and Mascis (2009) developed an optimal control system to manage the station's operations using a branch and bound algorithm. Liu and Kozan (2011) pointed out that scheduling of prioritized express trains to traverse continuously without interruption is a no-wait scheduling problem. Bakery production can also be modeled as no-wait scheduling problem. (Hecker, Stanke, Becker, and Hitzmann, 2014)

Extension of DPFSP with no-wait constraint was analyzed by Lin & Ying (2016) very first time. They solved the problem using mixed integer programming (MIP), mathematical model and an iterated cocktail greedy (ICG) algorithm. Komaki & Malakooti (2017) applied several heuristics to solve the problem from the literature which were used to deal with conventional no-wait flowshop. Due to NP-hard nature of the problem, a metaheuristic, general variable neighborhood search (GVNS) algorithm was also applied to solve the problem.

Shao et.al (2017) also studied no-wait DPFSP problem for makespan objective using iterated greedy algorithm. An improved NEH heuristic is proposed for generation of initial solution. Further, three variants of iterated greedy algorithm are used to solve the problem. Proposed results are compared with all the results published for the problem until then.

3.4 Heterogenous DPFSP

Most of the literature about DPFSP problem focuses on homogenous factories for processing of jobs whereas, in factual world heterogeneous machines are more common (Wen *et.al*, 2011. Khedr, 2011; KollOdziej & Khan, 2012). The difference between factories can be either in machine speed, technology, labour skills etc. These differences cause variation in achieved objective in different factories. In textile industry, spinning, which is a process of converting fibres into yarn, is done in different parallel spinning plants. These plants can have same spindle capacity and type of yarn manufactured but different spindle speed at the same time. This speed factor is needed to be kept in consideration while planning for allocation of jobs to either of the plant. Speed scaling impacts the processing speed, hence makespan of a job. This phenomenon of speed scaling is also considered for energy efficient scheduling (Ding et.al, 2016). A phenomenal work is done for speed scaling of flowshop environment for objective related to carbon emission efficiency and computer processing (Chau et.al, 2017). We didn’t find any paper related to speed scaling in DPFSP environment.

3.5 Research gaps

The reviewed literature indicates that distributed permutation flow shop is relatively new research area with scant available literature. DPFSP can be widely divided in two categories, homogenous and heterogenous. Most of the work is performed for homogenous factories. Very first published paper (Naderi & Ruiz, 2010) is focused on minimization of makespan in homogenous environment. Many of the subsequent papers deal with the same problem for makespan objective to compare and improve the results. Future work can be focused on DPFSP problem with constraints such as sequence dependent set up time, factory eligibility and machine blocking etc. Though most of the work performed is for identical factories whereas in real world factories may be unidentical. Future research can be extended to any of above unattended areas to make it more realistic. Keeping in view, current research deals with both homogenous and heterogenous problems. Heterogenous DPFSP problem is solved for an additional no-wait constraint.

CHAPTER 4

Homogenous Distributed permutation flowshop scheduling problem

This chapter considers the distributed permutation flowshop scheduling problem (DPFSP) which is an extension of permutation flowshop scheduling problem (PFSP). In DPFSP, there are multiple parallel factories instead of one factory as in PFSP. Each factory consists of same number of machines, and jobs can be processed in either of the factories to perform all necessary operations. This paper considers DPFSP for minimizing the total completion time objective. A MILP formulation is developed to find the optimal solution. To solve the problem, a metaheuristic, tabu search (TS) is proposed. Numerical experiments are performed on benchmark problem instances from the literature, and results of the proposed method are compared with current metaheuristics in the literature for this problem. The tabu search outperforms all existing metaheuristics in terms of solution quality.

4.1 Introduction

Scheduling deals with the allocation of resources, typically machines, to tasks (commonly referred to as jobs) over time with the goal of optimizing a given objective (Pinedo, 2012). Optimal efficiency can be achieved by scheduling jobs proficiently. In a conventional manufacturing plant, jobs are usually scheduled by some skilled persons using their self-developed rules. Extensive study on scheduling led researchers to develop different models and methods for specific production environments. The permutation flowshop problem (PFSP) is one of these manufacturing environments, where a series of operations are performed on every job in the same sequence. All jobs follow the same route, as the machines are placed in the same sequence. The PFSP is one of the widely studied problems in the area of scheduling.

According to three filed notations in scheduling, PFSP for the total completion time objective function can be denoted as $F/permu/\sum C_j$. A generalization of PFSP was introduced by Naderi and Ruiz (2010) to address the problem of multiple parallel flowshops. This problem is named as distributed permutation flowshop or DPFSP, because it distributes jobs to different factories. In DPFSP, there are multiple parallel factories, instead of one as in PFSP. Each factory consists of same number of machines, and any job can be processed in any of the factories to perform all necessary operations.

The need for DPFSP arises because economic globalization and frequently changing market structures force organizations to move to geographically scattered, decentralized plants instead of centralized plants. By exploiting the DPFSP environment, companies can reduce manufacturing costs and organizational risk and can improve the quality of products (Kahn et al, 2012). Due to new expectations and higher mandates in recent decades, an editorial was written in *International Journal of Production Research* by Hing and Sai (2013) about the optimization models used for solving DPFSP. This article mentions that market competitiveness and the demand for higher utilisation of resources can only be addressed by incorporating distributed manufacturing rather than centralized or single site in production fleets.

Distributed manufacturing is a common methodology currently being used in different industries including automotive industry (Gnoni et al, 2003), apparel industry (Leung et al, 2003), and steel industry (Sambasivan and Yahya, 2005). Sometimes, distributed manufacturing is performed in production facilities located in different parts of the continent (Wang et al, 2007;

Wilkinson et al, 1996). The DPFSP can be used in the above-mentioned industries because similar activities are performed in different places.

This paper considers DPFSP problem to minimize the total completion time (TCT) objective which is analysed by Fernandez et al, (2018). The job flow time represents the total time spent by the job in the system. The flow time of a job is associated with a cost factor known as inventory cost. Thus, minimizing total completion time of all jobs reduces the inventory cost through minimizing waiting time in the system. A mathematical model is developed to find the optimal solution. In addition, a metaheuristic, tabu search (TS) is proposed to solve the problem. An important feature of proposed tabu search is the use of an *Insertion Cost Matrix* and an improvement procedure. Introduction of an *Insertion Cost Matrix* helps to reduce CPU time of the proposed tabu search. An improvement procedure is used to bring solution to local optima of the current search region. Results of proposed tabu search is compared with existing metaheuristics.

The chapter is organized in 4 sections. Section 4.2 introduces the problem description and formulation, while section 4.3 presents the solution methods. Experimental results for total completion time objective are detailed in section 4.4. Finally, the chapter is concluded in section 4.5.

Reviewed literature indicates that DPFSP is attracting researchers' attention due to its application in the era of globalization. Because of globalization, a company may have different factories, and the same products can be produced in different factories and in different locations. Popularity of DPFSP is clear from the number of research papers published in recent years. It can be observed that though research is expanding, most of the research is focused on makespan objective function. Makespan represents the time when last job leaves the system, but makespan does not capture the performance of individual jobs. However, a cost factor in terms of inventory is associated with the flow time of the jobs in the system. This inventory may be in either raw material, work in process, or finished goods form. Cost related to the inventory can be reduced by focusing the flow time of individual jobs. Consequently, minimization of total completion time objective can be considered as a good measure to minimize the inventory cost. Therefore, we have considered the DPFSP with total completion time objective. For this purpose, a metaheuristic tabu search is proposed. A mathematical model adapted from Naderi and Ruiz (2010) has been developed to test the performance of developed methods by solving small sized problem instances. Naderi and Ruiz (2010) presented six MILP models in which the MILP model with a smaller number of variables and constraints is used in this chapter. Performance of proposed metaheuristic TS is compared with existing metaheuristics (BSIG, MIG, SS and EA) for DPFSP with a flow time objective.

4.2 Problem description and formulation

The DPFSP problem's setting consists of F identical parallel factories or flowshops. Each factory consists a set of M machines. There are n jobs ($j = 1,2,3, \dots n$) to be processed with M operations to be performed on each job to make a final product. Since each factory has M machines, a job can be processed in any factory. The M operations of jobs are performed on M machines of a factory. It takes p_{ji} time units to perform job j on machine i regardless of the factory. The processing time of a job doesn't change from factory to factory. A job can be assigned to any factory to perform all M operations. It is assumed that all operations of a job are performed in the same factory. Thus, once assigned, a job is not allowed to transfer to any other factory. It is assumed that all parameters are deterministic. Machine breakdown and set up times are negligible.

Each job can be processed at only one machine at a certain given time, and each machine can process only a single job at any specified time interval. At the same time, a job cannot be interrupted once it starts processing. This process involves determining the allocation of jobs to each factory and their sequence in order to minimize the total completion time objective. The following notations are used to represent the problem mathematically.

n	Number of jobs
m	Number of machines in each factory
F	Number of factories
j,k	index for jobs, $j,k \in \{1, 2, \dots, n\}$
k	job position in the sequence, $k \in \{1, 2, \dots, n\}$
f	index for factories, $f \in \{1, 2, \dots, F\}$
i	index for machines, $i \in \{1, 2, \dots, m\}$
P_{ji}	processing time of job j at machine i
$C_j(\pi)$	Completion time of job j for sequence π (Reviewer Suggested to remove)
M	A sufficiently large positive number

The solution determines the allocation of jobs and their processing sequence σ^f in each factory f, where σ is the sequence of jobs $\pi = [\pi^1, \pi^2 \dots \pi^F]$. Expression for allocated jobs to factory 1 can be shown by $\pi^1 = [\pi^1(1), \pi^1(2), \dots, \pi^1(n_1)]$, where n_1 is the number of jobs allocated to factory 1. The objective function is to minimize the total completion time of the jobs. The total completion time for solution π is $C(\pi) = [C^1(\pi), C^2(\pi), \dots, C^F(\pi)]$, where $C^f(\pi)$ represents the total completion time of all the jobs at factory f. The mathematical model for minimizing the total completion time is adopted from the mathematical model proposed by Naderi and Ruiz (2010) to where the only difference comes from the objective function which is to minimize the makespan. The selected model has $n(n+1)$ number of binary variables, nm number of continuous variables and $3n + 2 + nm(1 + n) + 12n(n - 1)$ number of constraints (NC). The MILP model uses following decision variables:

$C_{k,i}$ Continuous variable representing completion time of the job in position k on machine i.

$X_{k,j}$ Binary variable that takes value 1 if job j is processed immediately after job k and 0 otherwise.

$$Min \sum_{k=1}^{n} C_{km} \quad (1)$$

Subject to:

$$\sum_{k=0}^{n} X_{kj} = 1 \quad j \in \{1, 2, \dots, n\} \quad (2)$$

$$\sum_{j=0}^{n} X_{kj} \leq 1 \quad (3)$$

$$\sum_{j=1}^{n} X_{0j} = F \quad (4)$$

$$\sum_{k=1}^{n} X_{ko} = F - 1 \quad (5)$$

$$X_{kj} + X_{jk} \leq 1 \quad \forall k \in \{1, 2 \dots n\text{-}1\}\ j > k \quad (6)$$

$$C_{j,i} \geq C_{j,i-1} + p_{j,i} \quad j \in \{1, 2, \dots, n\} \quad (7)$$

$$C_{j,i} \geq C_{k,i} + p_{j,i} + (X_{k,j} - 1).M \quad \forall j, k, i, k \neq j \quad (8)$$

$$C_{k,0} = 0 \quad \forall k \quad (9)$$

$$C_{k,i} \geq 0 \quad \forall k,i \quad (10)$$

$$X_{kj} \in \{0,1\} \quad \forall_{k,j,j \neq k} \quad (11)$$

Equation (1) represents the objective function which is the total completion time. Constraint set (2) promises that every job can be at only one position and in only one factory at a time. It is assured by constraint set (3) that there is a maximum one succeeding job for every job in the sequence. Constraint set (4) ensures that dummy job 0 appears F times in the sequence as a predecessor. Constraint set (5) assures dummy job 0 must be a successor F–1 time. Constraint set (6) assures that a job cannot be both a successor and predecessor of a job at any given time. Constraint set (7) ensures that processing of a job on specific machine cannot start before completion of processing of same job on the previous machine. This constraint set assures that each job is processed by only one machine in each time window. Constraint set (8) makes sure that a job cannot start processing on a machine before completion of processing of previous job on the same machine. It specifies that at any given time a machine can process only one job. The constraint sets (10) and (11) represent the decision variables.

4.3 Solution method

To solve the DPFSP, we proposed a TS metaheuristic. The TS was originally proposed by Glover (1986) to deal with combinatorial problems. Little work is performed to solve DPFSP problem using tabu search (Gao et al, 2013), though it is used widely to solve flowshop scheduling problem. The TS starts with an initial solution and dynamically moves from one solution to another by visiting neighbourhood solution. The neighbourhood solution is generated through the movement of jobs from one position to another. This movement is called *move*. Nodes of these recent moves are saved in a short-term memory known as a *tabu list* to avoid cycling. These moves are prohibited from getting reversed, unless a defined number of iterations are passed. The number of these iterations is known as *tabu tenure*. This procedure is continued until stopping criteria is reached. Pseudocode of the TS is presented in Figure 4.

Generate an initial solution π using construction heuristic
Initialize best solution π* ← π
Initialize *InsertionCostMatrix* A^{Net} (π)
    Set Counter=0
While termination criteria not met **do**
    Counter ++
    Generate neighbourhood solutions *N(π)*
    Choose the best neighbourhood solution which is not tabu
    π'ϵN(π)
    Set π=π'
    Update *InsertionCostMatrix* A^{Net} (π)
    if Counter = β
        π ← Improvement scheme (π)
        Initialize *InsertionCostMatrix* A^{Net} (π)
      Counter = 0
    end if
    if π is better than π*
        π* = π
    end if
end while
return π*

Figure 4 Pseudocode of tabu search

4.3.1. Generating initial solutions

A construction heuristic is also known as an augmentation heuristic. This is one of the fastest ways to reach a feasible solution for a specific problem. Pseudocode for a construction heuristic is summarized in Figure 5.

Procedure Construction heuristic
//Job insertion order

Ω =Jobs stored randomly *($\Omega = \{w_1, w_2, \ldots, w_n\}$*
$\pi^f = \varphi, \forall f$
for k = 1 to *n* do
    $(l, f) \leftarrow$ *BestInsertionCost* (w_k, π)
    $\pi^f \leftarrow$ Update sequence of factory *f* by inserting w_k in position *l* of partial sequence π_f
end for
end procedure
return π

Figure 5 Pseudocode for construction heuristic

In this heuristic, jobs are selected randomly one-by-one and inserted iteratively in the partially built schedule. The procedure starts with a null partial sequence. The cheapest insertion method is used to insert jobs in a partially built sequence using *Best Insertion Cost* procedure, as shown in Figure 6. The resulting sequence is used as a current partial sequence of *k*-jobs for next iteration. The process stops when all jobs are inserted in partial sequence π. Pseudocode for best insertion cost procedure is presented in Figure 6.

Procedure *BestInsertionCost* (w_k, π)
$C^{T}(\pi)$ = Total completion time of partial sequence π
$C^{Best} = \infty$
for f = 1 to F do
    for k=1 to nf + 1do
        $\overline{\pi}^f$ = Sequence found by inserting job w_k in position k of partial sequence π^f
        $\overline{C}^f(\overline{\pi}^f) \leftarrow$ Total completion time of sequence $\overline{\pi}^f$ for factory *f*
        if $(C^{T}(\pi) + \overline{C}^f(\overline{\pi}^f) - C^f(\pi^f)) \leq C^{BV}$
        $l = k$
        $f = f$
        end if
    end for
end for
Report (l , f)
end procedure

Figure 6 Pseudocode for best insertion cost procedure

4.3.2. Initialize insertion cost matrix

CPU time is one of the important criteria for effectiveness of an algorithm. In the proposed TS, a move is defined as the movement of a job to the best position in another or in same factory. In a TS, all neighbourhood moves are explored to find the next solution, but only one move is accepted. In the next iteration, again, all the neighbourhood solutions are evaluated. Most of the evaluated neighbourhood moves are repeated from one iteration to another and only one move is accepted in a given iteration. A repeated calculation of neighborhood moves can be avoided, if a suitable database is maintained. In this paper, we propose *Insertion Cost Matrix* to avoid repetitive calculation.

The proposed database calculates the change in flow time for removal of a job from a factory or addition of a job in a factory. Our proposed database is based on the change in total completion time of a factory. Flow time of a factory is defined as total completion time of all jobs in the factory. The database keeps track of minimum increase in flow time of a factory, if a job is inserted into a factory. Similarly, it also keeps track of decrease in flow time of a factory, if a job is removed from it. Proposed *Insertion Cost Matrix* uses following notations.

Notation	**Short Definition**
δ_{jf}^{+}	Change in flow time of factory *f* due to movement of job *j* from its current factory to factory *f*
δ_{j}^{-}	Change in flow time due to removal of a job from its current factory
δ_{jf}^{Net}	Net change in flow time of a solution due to movement of a job *j* to factory *f*
A^{+}	Positive cost matrix for increase in flow time
A^{-}	Negative cost matrix for decrease in flow time of factory
A^{Net}	Net cost matrix for increase in flow time of solution

Figure 7 Notations for insertion cost matrix

The value of δ_{jf}^{Net} can be calculated by adding δ_{jf}^{+} and δ_{j}^{-} i.e. $\delta_{jf}^{Net} = \delta_{jf}^{+} + \delta_{j}^{-}$. In all these definitions, change is defined as the difference between new value and old value (i.e. new value – old value). Following three matrices (Figure 8) can be constructed to represent the increase or decrease of flow time of all jobs in all factories.

$$A^{+} = \begin{bmatrix} \delta_{11}^{+} & \delta_{12}^{+} & \cdots & \delta_{1f}^{+} \\ \delta_{21}^{+} & \delta_{22}^{+} & \ldots & \delta_{2f}^{+} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1}^{+} & \delta_{n2}^{+} & \ldots & \delta_{nf}^{+} \end{bmatrix} \qquad A^{-} = \begin{bmatrix} \delta_{1}^{-} \\ \delta_{2}^{-} \\ \vdots \\ \delta_{n}^{-} \end{bmatrix}$$

$$A^{Net} = \begin{bmatrix} \delta_{11}^{net} & \delta_{12}^{net} & \cdots & \delta_{1f}^{net} \\ \delta_{21}^{net} & \delta_{22}^{net} & \ldots & \delta_{2f}^{net} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1}^{net} & \delta_{n2}^{net} & \ldots & \delta_{nf}^{net} \end{bmatrix}$$

Figure 8 Development of insertion cost matrix

The *Insertion Cost Matrix* requires a calculation of change in objective function for insertion operation. This insertion operation is expedited by introduction of a speedup mechanism to reduce the computational effort. Computational complexity to calculate flow time for inserting a removed job from a factory is $O(nm)$ which results in complexity of $O(n^2m)$ for a single step (Fernandez et al, 2018). Level of this complexity can be reduced significantly by reducing the number of calculations to be performed for every movement of a job. Consider an example of a factory f with five (5) jobs S_0= {1, 2, 3, 4, 5}. Consider the insertion of job 6 in the sequence of factory f. Insertion of job 6 will result in six possible sequences as; S_1= {1, 2, 3, 4, 5, 6}, S_2 = {1, 2, 3, 4, 6, 5}, S_3= {1, 2, 3, 6, 4, 5}, S_4= {1, 2, 6, 3, 4, 5}, S_5={1, 6, 2, 3, 4, 5} and S_6={6, 1, 2, 3, 4, 5}. For schedule S_1, first five jobs are in the same sequence as in S_0, which would require no recalculation of flow times of jobs {1, 2, 3, 4, 5}. Flow time can be obtained from the sequencing of S_0. Only completion time of job 6 needs to be calculated. For schedule S_2 first four jobs are in the same sequence as in S_0 which helps in avoiding a re-recalculation of flow times of jobs {1, 2, 3, 4}. Same is the case for sequence S_3, S_4 and S_5 where we do not need to recalculate flow time of jobs {1, 2, 3}, {1, 2} and {1}. Process of using pre-calculated flow time results in reduction of computational complexity to $O(n^2m\text{-}((n^2\text{-}3n+2)/2)m)$.

4.3.3. Generating neighborhood solutions

In this neighborhood search process, current sequence (π) is applied with moves to generate set of neighborhood solutions. A move in the proposed TS considers the move of a job from its current factory to the best position in either a different or same. The total number of neighborhood moves will be $n * f$ and are represented as $N(\pi)$. Cost of all these moves are already stored in the net cost matrix A^{Net}. Hence, the net cost matrix A^{Net} can be scanned to find the best neighborhood move. While scanning net cost matrix A^{Net}, only non tabu moves are considered to find best neighborhood moves. Finally, best neighbor sequence π' becomes current solution for next iteration.

4.3.4. Updating insertion cost matrix

After every iteration, all three (positive, negative and net cost) matrices of *Insertion Cost Matrix* are updated and are made available for the next iteration based on the previous move. Consider movement of job j from factory f_1 to factory f_2. In case of a positive cost matrix (A^+), the columns related to factory f_1 and f_2 are updated, while columns for all other factories remain same. For a negative cost matrix (A^-), flow time of all the jobs from factory f_1 and f_2 are updated. Finally, net cost matrix (A^{Net}) is updated accordingly. Consider an instance of 8 jobs and 4 factories. Consider current solution π = {(1,5) (2,6) (3,7) (4,8)}.

Current Sequence

$\pi^1 =$ 1, 5 $\quad\quad$ $\pi^3 =$ 3, 7

$\pi^2 =$ 2, 6 $\quad\quad$ $\pi^4 =$ 4, 8

If the neighborhood accepts movement of job 6 from factory 2 to factory 3, the updated sequence will be;

Updated Sequence

$\pi^1 =$ 1, 5 $\quad\quad$ $\pi^3 =$ 3, 7, 6

$\pi^2 =$ 2 $\quad\quad$ $\pi^4 =$ 4, 8

The recalculation of cells required in matrix A^+ and A^- are shown using rectangular boxes here in Figure 9.

$$A^+ = \begin{bmatrix} \delta_{11}^+ & \delta_{12}^+ & \delta_{13}^+ & \delta_{14}^+ \\ \delta_{21}^+ & \delta_{22}^+ & \delta_{23}^+ & \delta_{24}^+ \\ \delta_{31}^+ & \delta_{32}^+ & \delta_{33}^+ & \delta_{34}^+ \\ \delta_{41}^+ & \delta_{42}^+ & \delta_{43}^+ & \delta_{44}^+ \\ \delta_{51}^+ & \delta_{52}^+ & \delta_{53}^+ & \delta_{54}^+ \\ \delta_{61}^+ & \delta_{62}^+ & \delta_{63}^+ & \delta_{64}^+ \\ \delta_{71}^+ & \delta_{72}^+ & \delta_{73}^+ & \delta_{74}^+ \\ \delta_{81}^+ & \delta_{82}^+ & \delta_{83}^+ & \delta_{84}^+ \end{bmatrix} \qquad A^- = \begin{bmatrix} \delta_1^- \\ \boxed{\delta_2^-} \\ \boxed{\delta_3^-} \\ \delta_4^- \\ \delta_5^- \\ \boxed{\delta_6^-} \\ \boxed{\delta_7^-} \\ \delta_8^- \end{bmatrix}$$

Figure 9 Example of updating insertion cost matrix

4.3.5. Improvement scheme

One of the new features of proposed TS approach is the adoption of a solution improvement scheme. In every β iterations, the current solution is improved. Purpose of solution improvement scheme is to bring the solution back to current local optima of current search region. Solution improvement scheme is presented in Figure 10, which is an adaptation of adaptive large neighborhood search (ALNS) from Ropke and Pisinger (2006). The ALNS has popularly been used for solving many variants of vehicle routing problem (VRP). It is an iterative process of a destroy and repair operator of existing solution. The destroy operator removes a part of current solution, and the repair function puts that part back into the same solution which results in a different solution. The whole procedure is repeated 10 times. Best solution found in 10 iterations is finally improved using local search schemes.

Procedure Improvement Scheme (π)
Initialize best solution π* as π
While termination criteria not met do
    {randomly draw number of jobs σ for removal}
    σ ← draw number of jobs to remove
    {apply randomly selected destroy operator with probability of ω^-
    π’ ← ApplyDestroy (π)
    {apply a repair operator}
    π’← ApplyRepair (π’)
    If π is better than π*
        π*=π
    end if
end while
{apply local search, intra and inter factory moves are applied}
π ← Apply Local Search (π*)
end procedure

Figure 10 Pseudocode of the improvement scheme

Initially, the number of jobs to be removed (σ) are selected randomly in the range of [0.1n to 0.4n]. Then, removal operator is used to remove σ number of jobs from the current solution. We used three types of removal operators: 1) Random removal operator, 2) Reduction cost removal operator, and 3) Relocation cost removal operator. One of the removal operators is selected randomly in each iteration. The reduction cost removal operator removes first σ jobs with lowest values in reduction of costs (i.e. δ_i^-). Reduction cost of a job is defined as a reduction in flow time of a solution, if the job is removed from its original position. The relocation cost removal operator removes first σ jobs with lowest value of relocation cost (i.e. the value of δ_{ij}^-). Relocation cost of a job is defined as a reduction in flow time if a job is removed from its original position and reinserted in the best position of the solution.

We use three types of repair operators 1) Greedy insertion repair operator 2) Regret insertion operator 3) Grasp insertion operator. Greedy operator iteratively performs best possible insertion from the list of unassigned jobs. Idea of regret insertion is to anticipate the future effect of an insertion operation. Regret value of each unassigned job is calculated as the difference between insertion cost at the best position and second-best position. The unassigned job with maximum regret value is selected for insertion. Grasp insertion is a variation of greedy insertion. In this operation, increase in cost for the insertion of all unassigned jobs is calculated and is stored in the list of size *D* in ascending order of cost increase. Instead of selecting the best job for insertion, a random job from the first *rand***D* jobs is selected where *rand* denotes degree of randomness taking value between 0 and 0.2. The parameters used in the scheme are decided by performing sensitivity analysis with limited CPU time.

4.3.6. Local search

A simple iterated local search is applied to the solution after performing a destroy and repair operation 10 times. In this local search, all jobs are first stored randomly in a set Ω. The jobs are then selected one-by-one, for possible relocation in either same or a different factory. If an improved solution is found, the sequence is immediately updated for evaluation of relocation of next job. The whole process is repeated until a solution keeps on improving. Pseudocode for local search is provided in Figure 11. The procedure requires finding best relocation place *l* in best relocation factory *f* for job w_k. The *insertion cost matrix* A^{Net} can be used to find the best value of *l* and *f*.

Procedure Local Search (π)
do
$C^0 \leftarrow$ Initial total completion time of sequence
$\Omega \leftarrow$ Jobs stored randomly $(\Omega)=\{w_1, w_2, \ldots, w_k\}$
for k = 1 to n do
    $(l, f) \leftarrow$ Best Relocate Cost (w_k, π)
    $\pi^f \leftarrow$ update sequence of factory *f* by inserting w_k in position l of partial sequence π_f
end for
$C \leftarrow$ total completion time of sequence π
while $(C < C^0)$
end procedure

Figure 11 Pseudocode of the iterated local search procedure

4.3.7. Tabu list and tabu tenure

Tabu search is named after its ability to make some movements of jobs 'tabu' in order to prevent cycling. These moves are not allowed to reverse until iterations are defined by tabu tenure λ. A short-term memory, known as a tabu list, is used to save the properties of prohibited solutions visited previously. Storing a list of all visited solutions is an expensive process; therefore, only some of the attributes are saved. A tabu list saves the attributes of moves relevant to neighborhood. We use term T_{if} for the tabu list which represents number of iterations until movement of job *i* to factory *f* is prohibited. Consider a best neighborhood move in which job *j* is moved to factory f_2 from factory f_3 in iteration *t*. This movement can be stored so as to not allow reverse movement of

j to factory f_2 for the next tabu tenure (λ) iterations. Thus, the value of T_{if} will be reset to $T_{\mathrm{ij}} + \lambda$ (*i.e.* $\mathrm{t} + \lambda$)

4.4 Numerical experiments

In this section, we discuss benchmark problem instances in the experiments; we also show and analyze the results of proposed solution methods.

4.4.1. Benchmarking instances

The benchmark data set provided by Naderi and Ruiz (2010) is used for comparison and available at http://soa.iti.es. They provide two types of data sets: small-sized instances and large-sized instances. Small-sized instances are used to compare the solution of proposed solution methods with optimal solution obtained with application of mathematical model. Large-sized instances are used to compare the relative performance of proposed solution methods.

In small-sized instances, we used {4, 6, 8, 10} jobs at {2, 3, 4, 5} machines. The number of factories is 3 for all problem instances. Each combination of jobs, machines and factories has 5 instances. Thus, a total of 80 instances are considered in small-sized instances. In large-sized instances, there are {20, 50, 100, 200, 500} jobs at {5, 10, 20} machines and in {2, 3, 4, 5, 6, 7} factories. Each combination of jobs, machines and factories has 10 instances. Thus, the large problem data set has a total of 720 problem instances.

4.4.2. Computational results

Proposed solution methods are coded using C language to solve the problem instances. Tabu search is run for 10,000 iterations, and CPU time is calculated accordingly. Tabu tenure λ is set to 10. Experiments are run on a server with four 2.1 GHz processors having 16-core each and 256 Gb of RAM memory. Problems of small size instances are solved by using MILP with CPLEX solver (version 12.6). The AMPL is running on a iMac desktop with 3.3GHz and 8 GB RAM. We have used the following abbreviations for reporting the results:

OPT	Optimal solution using solver CPLEX
CPU	Processing time
TCT	Total completion time
TS	Tabu search
MIG	Modified iterated greedy algorithm of Fernandez Viagas and Framinan (2015)
BISG	Bounded search iterated greedy algorithm of Lin et al. (2013)
SS	Scatter search of Naderi and Ruiz (2014)
EA	Evolutionary algorithm of Fernandez et al. (2018)
APD	Absolute percentage deviation

Performance of proposed methods for total completion time is evaluated using absolute percentage deviation.

$$APD_j^i = \{(H_j^i - O_i)\ /\ O_i\} * 100$$

where

APD_j^i = Absolute percentage deviation of the algorithm *j* from the CPLEX for problem instance *i*

H_j^i = Total completion time obtained by algorithm *j* for problem instance *i*

O_i = Total completion time obtained by the CPLEX for problem instance *i*

4.4.2.1 Computational results for small-sized instances

Results of small-sized instances are compared with the optimal solution obtained by solving MILP using CPLEX solver, since the solution of the existing algorithms is not available for this data set. Therefore, the proposed TS is only compared with the optimal solution.

Table 3 indicates that TS provided the best results for 97.5% of 80 instances. For small-sized instances, TS proved to be an effective metaheuristic with an average APD of 0.005%. In case of CPLEX, the CPU times for 4, 6 and 8 jobs are less than 1 second except for one instance. For 10 jobs, it increases and fluctuates between 5 to 123 seconds with an average of 16.82 seconds per instance. The CPU time for TS is less than 1 second. It can be observed from above analysis that tabu search provided best results with lowest value of APD as compared to CPLEX. Moreover, the TS maintained shortest CPU time compared to CPLEX. Construction heuristic is used as an initial solution for tabu search. Tabu search improved initial solution by 1.485% (1.49-0.0047) on average for small sized instances.

Table 3 TCT of CPLEX and TS for small-sized instances

No	**Instance**	**CPLEX**		**Construction Heuristic**		**Tabu Search**	
		TCT	CPU	TCT	APD	TCT	APD
1	I_3_4_2_1	391	<1	391	0.00	391	0.00
2	I_3_4_2_2	542	<1	543	0.18	543	0.18
3	I_3_4_2_3	555	<1	561	1.08	555	0.00
4	I_3_4_2_4	607	<1	607	0.00	607	0.00
5	I_3_4_2_5	440	<1	440	0.00	440	0.00
6	I_3_4_3_1	523	<1	549	4.97	523	0.00
7	I_3_4_3_2	515	<1	515	0.00	515	0.00
8	I_3_4_3_3	695	<1	695	0.00	695	0.00
9	I_3_4_3_4	570	<1	570	0.00	570	0.00
10	I_3_4_3_5	637	<1	689	8.16	637	0.00
11	I_3_4_4_1	877	<1	877	0.00	877	0.00
12	I_3_4_4_2	852	<1	852	0.00	852	0.00
13	I_3_4_4_3	733	<1	733	0.00	733	0.00
14	I_3_4_4_4	1031	<1	1031	0.00	1031	0.00

15	I_3_4_4_5	790	<1	848	7.34	790	0.00
16	I_3_4_5_1	1177	<1	1177	0.00	1177	0.00
17	I_3_4_5_2	1088	<1	1088	0.00	1088	0.00
18	I_3_4_5_3	1053	<1	1082	2.75	1053	0.00
19	I_3_4_5_4	932	<1	932	0.00	932	0.00
20	I_3_4_5_5	867	<1	872	0.58	867	0.00
21	I_3_6_2_1	719	<1	719	0.00	719	0.00
22	I_3_6_2_2	613	<1	613	0.00	613	0.00
23	I_3_6_2_3	914	<1	930	1.75	914	0.00
24	I_3_6_2_4	583	<1	600	2.92	583	0.00
25	I_3_6_2_5	773	<1	781	1.03	773	0.00
26	I_3_6_3_1	852	<1	852	0.00	852	0.00
27	I_3_6_3_2	961	<1	961	0.00	961	0.00
28	I_3_6_3_3	721	<1	722	0.14	721	0.00
29	I_3_6_3_4	1051	<1	1051	0.00	1051	0.00
30	I_3_6_3_5	1234	<1	1238	0.32	1234	0.00
31	I_3_6_4_1	1272	<1	1286	1.10	1272	0.00
32	I_3_6_4_2	1284	<1	1316	2.49	1284	0.00
33	I_3_6_4_3	1456	<1	1488	2.20	1456	0.00
34	I_3_6_4_4	1598	<1	1598	0.00	1598	0.00
35	I_3_6_4_5	1604	<1	1607	0.19	1604	0.00
36	I_3_6_5_1	1672	<1	1672	0.00	1672	0.00
37	I_3_6_5_2	1400	<1	1414	1.00	1400	0.00
38	I_3_6_5_3	1857	<1	1891	1.83	1857	0.00
39	I_3_6_5_4	1849	<1	1849	0.00	1849	0.00
40	I_3_6_5_5	1189	<1	1189	0.00	1189	0.00
41	I_3_8_2_1	1140	<1	1180	3.51	1140	0.00
42	I_3_8_2_2	1038	<1	1062	2.31	1038	0.00
43	I_3_8_2_3	892	<1	899	0.78	892	0.00
44	I_3_8_2_4	1146	<1	1149	0.26	1146	0.00
45	I_3_8_2_5	1015	<1	1028	1.28	1015	0.00
46	I_3_8_3_1	1442	<1	1477	2.43	1442	0.00
47	I_3_8_3_2	1412	<1	1443	2.20	1412	0.00
48	I_3_8_3_3	1305	<1	1324	1.46	1305	0.00
49	I_3_8_3_4	1099	<1	1167	6.19	1099	0.00
50	I_3_8_3_5	1390	<1	1393	0.22	1390	0.00
51	I_3_8_4_1	1808	<1	1850	2.32	1808	0.00
52	I_3_8_4_2	1792	<1	1814	1.23	1792	0.00
53	I_3_8_4_3	1827	<1	1853	1.42	1827	0.00
54	I_3_8_4_4	1904	<1	1973	3.62	1904	0.00
55	I_3_8_4_5	1878	<1	1894	0.85	1878	0.00
56	I_3_8_5_1	2218	<1	2283	2.93	2218	0.00

57	I_3_8_5_2	2112	1.03	2174	2.94	2116	0.19
58	I_3_8_5_3	2179	<1	2241	2.85	2179	0.00
59	I_3_8_5_4	2422	<1	2424	0.08	2422	0.00
60	I_3_8_5_5	2345	<1	2375	1.28	2345	0.00
61	I_3_10_2_1	1229	4.6	1259	2.44	1229	0.00
62	I_3_10_2_2	1308	4.33	1316	0.61	1308	0.00
63	I_3_10_2_3	1442	13.73	1461	1.32	1442	0.00
64	I_3_10_2_4	1466	10.85	1488	1.50	1466	0.00
65	I_3_10_2_5	1115	4.77	1135	1.79	1115	0.00
66	I_3_10_3_1	2022	123.02	2042	0.99	2022	0.00
67	I_3_10_3_2	2344	21.22	2405	2.60	2344	0.00
68	I_3_10_3_3	1849	12.75	1909	3.24	1849	0.00
69	I_3_10_3_4	1864	5.17	1932	3.65	1864	0.00
70	I_3_10_3_5	1892	7.78	1937	2.38	1892	0.00
71	I_3_10_4_1	2464	34.08	2464	0.00	2464	0.00
72	I_3_10_4_2	2370	3.65	2411	1.73	2370	0.00
73	I_3_10_4_3	2360	1.8	2446	3.64	2360	0.00
74	I_3_10_4_4	2703	12.63	2741	1.41	2703	0.00
75	I_3_10_4_5	2641	5.54	2646	0.19	2641	0.00
76	I_3_10_5_1	2627	8	2672	1.71	2627	0.00
77	I_3_10_5_2	2716	33	2810	3.46	2716	0.00
78	I_3_10_5_3	3225	15.84	3337	3.47	3225	0.00
79	I_3_10_5_4	3072	5.6	3075	0.10	3072	0.00
80	I_3_10_5_5	2743	8.01	2823	2.92	2743	0.00
Average		1428.663	16.067	1451.76	1.49	1428.73	0.0047

4.4.2.2 Computational results for large-sized instances

Fernandez et al. (2018) used an evolutionary algorithm (EA) to compare the results with efficient metaheuristic methods applied for $DF \mid prmu \mid \sum C_j$ problems. They have compared the results of a modified iterated greedy algorithm (Fernandez-Viagas and Framinan, 2015), a bounded-search iterated greedy algorithm, (Lin et al. 2013) and a scatter search (Naderi and Ruiz, 2014). For comparison purpose, all algorithms were coded and run using same computer for the same stopping criteria.

Since optimal solution for large-sized instances for considered problem is available as per publication by Fernandez et al. (2018), relative percentage deviation is calculated with respect to best-known solution using following equation:

$$RPD_j^i = \{(H_j^i - B_i) \,/\, B_i\} * 100$$

where

RPD_j^i = Relative percentage deviation of algorithm *j* from best-known solution for problem instance *i*

H_j^i = Total completion time obtained by algorithm *j* for problem instance *i*

B_i = Total completion time obtained by best-known solution provided by Fernandez et al. (2018) for problem instance *i*

Results of proposed solution method are presented in Table 4 and 5 for stopping criteria *t·n·m·F*, where *t* is a time parameter. They (Fernandez et al, 2018) report the solution for different parameter values of *t* (*t* =0.5, *t* =1 and *t* =2). We used the solution obtained when time parameter *t* is set to 2. Therefore, all existing algorithms have same CPU time and are shown under only one column in Table 4. This setting of stopping criteria provides the best solution. Large-sized problems have 720 instances divided into twelve groups, based on combination of number of jobs and number of machines. Average of relative percentage deviation is grouped by *n* x *m*. Each group has sixty (60) problem instances. Table 4 presents relative performance of a proposed solution method in comparison to the state-of-the-art for flow time objective. Average RPD of TS is negative where negative value of average RPD shows that TS has improved the best-known existing solution by 0.094%. It can also be noticed that tabu search improved the initial solution by 2.83% (2.664+0.167) for large sized instances.

Table 4 Average RPD of the algorithms grouped by n and m

Group	n x m	Existing Algorithms					Tabu Search			
		TCT				CPU	Initial Solution		Final Solution	
		MIG	BSIG	SS	EA		TCT	CPU	TCT	CPU
G1	20 × 5	0.054	0.257	0.232	0.021	4.50	2.434	0.00	-0.007	0.36
G2	20 × 10	0.065	0.257	0.149	0.009	9.00	2.263	0.00	-0.005	0.79
G3	20 × 20	0.031	0.178	0.107	0.005	18.00	1.514	0.00	-0.001	1.57
G4	50 × 5	0.347	1.057	0.545	0.032	11.25	3.834	0.00	-0.070	5.75
G5	50 × 10	0.594	0.916	0.536	0.002	22.50	3.163	0.00	-0.075	11.63
G6	50 × 20	0.491	0.658	0.408	0.012	45.00	2.549	0.00	-0.024	22.18
G7	100 × 5	0.567	1.017	0.11	0.094	22.50	3.589	0.00	-0.107	49.40
G8	100 × 10	0.875	1.064	0.21	0.059	45.00	3.431	0.00	-0.144	100.27
G9	100 × 20	0.901	0.915	0.266	0.039	90.00	2.662	0.00	-0.130	202.01
G10	200 × 10	1.114	0.818	0.01	0.229	90.00	2.766	0.00	-0.290	1032.61
G11	200 × 20	0.978	0.745	0.021	0.11	180.00	2.260	0.00	-0.358	2104.20
G12	500 × 20	1.051	0.987	0.165	0.066	450.00	1.502	1.97	-0.789	54481.68
Average		0.589	0.739	0.23	0.056	82.31	2.664	0.16	-0.167	4834.37

Results reported in Table 4 show that the average RPD values of MIG, BSIG, EA and TS are 0.589, 0.739, 0.23, 0.056 and -0.167%, respectively. Results further indicate that the objective function values of TS are less than all four metaheuristics for $t \cdot n \cdot m \cdot F$ stopping criteria (t=2). The objective function values of MIG, BSIG, SS and EA are away from TS by 0.756%, 0.906%, 0.397% and 0.223%, respectively. It can be observed that TS performs better in terms of solution quality as well as in terms of CPU time for instances with 50 or less jobs. When number of jobs is more than 50, tabu search performs better in terms of solution quality; however, the CPU is higher than existing algorithms.

Results of different algorithms on the basis of number of factories are compared in Table 5. Behaviour of different metaheuristics for an increase in number of factories is presented in

Figure 12. It can be noticed that the performance of MIG and EA improves with an increase in number of factories. The behaviour is opposite in case of BSIG, SS and TS where their performance decreases, as number of factories increases.

Average RPD values of MIG varies from 0.392% to 0.890%, and BSIG varies from 0.575% to 0.903%. Similarly, average RPD values of SS varies from 0.162% to 0.295%, and for EA, it varies from 0.028% to 0.078%. The upward performance trend of TS indicates that RPD value increases with the increase in the problem as illustrated in Figure 12. In other words, the performance of the tabu search decreases with an increase in number of factories. Overall, tabu search provides improved results for 412 (57.22%) instances, poor results for 159 (22.08%) instances and same results for 149 (20.69%) instances as compared to the best-known solutions.

Table 5 Average RPD of the algorithms grouped by *F*

F	**Total Completion Time**					
	Existing Algorithms				Tabu Search	
	MIG	BSIG	SS	EA	Initial Solution	Final Solution
2	0.890	0.575	0.172	0.063	2.920	-0.378
3	0.712	0.707	0.162	0.078	2.710	-0.207
4	0.578	0.717	0.206	0.072	2.648	-0.147
5	0.513	0.903	0.258	0.055	2.691	-0.099
6	0.449	0.803	0.285	0.044	2.545	-0.103
7	0.392	0.728	0.295	0.028	2.469	-0.066
Average	0.589	0.739	0.230	0.056	2.664	-0.167

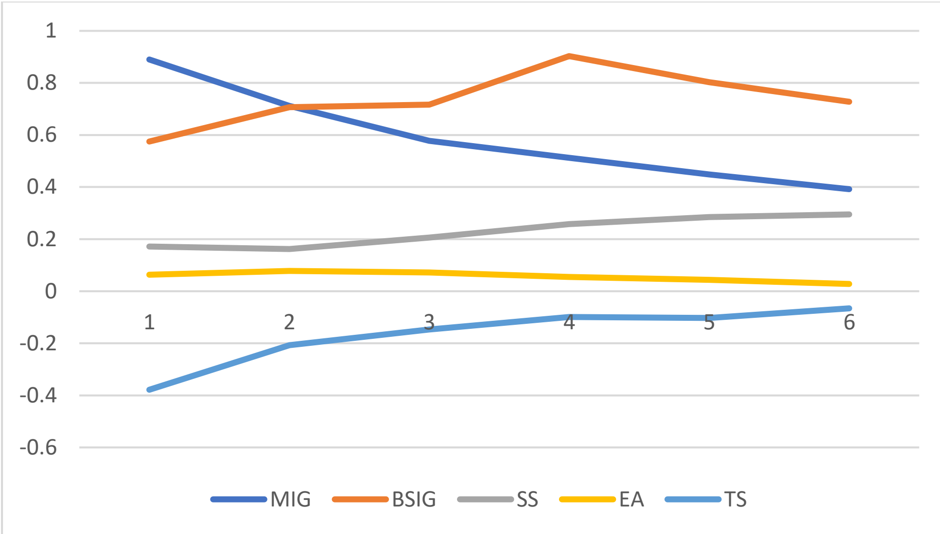

Figure 12 Comparative analysis of tabu search

4.5 Conclusion

This paper considers DPFSP problem with the objective of minimizing total completion time. The DPFSP involves processing of jobs through one of the different distributed flowshops (Fan, 2010) in contrast to a conventional flowshop, where there is only one factory. We propose a metaheuristic TS to solve the problem. The proposed TS uses two distinct features: a cost matrix and an improvement scheme. The cost matrix helps to reduce the CPU time, while the improvement scheme helps to improve the solution quality. Two types of already available data instances (small sized and large sized) are used to perform the analysis. The construction heuristic is used to develop an initial solution, which is later fed to TS for further processing. The proposed TS is compared with the existing algorithm. The TS outperformed all of the solution methods (heuristics and metaheuristics) in terms of solution quality. The TS provided an average of 0.167 % better results improving 412 (57.22%) instances out of 720. It can be established that TS resulted in a better solution quality. With above mentioned improvements, this paper provides new state-of-the-art results for the DPFSP problem.

Future work can be extended for consideration of the heterogeneous nature of factories. In a heterogenous factory, the process time of a job can vary from factory-to-factory in the same stage or machine. Also, research can be focused on due-date related objective functions. The due-date related objectives are relevant for a globally competitive manufacturing environment. Another extension of the conventional DPFSP Problem can be a factory eligibility constraint. This environment limits the allocation of jobs based on eligibility of processing in a certain factory. A job cannot be assigned to a factory where a job cannot be processed by at least a single machine of the factory.

CHAPTER 5

Heterogeneous no-wait distributed permutation flowshop scheduling problem

This chapter deals with solving distributed permutation flowshop scheduling problem for no-wait heterogenous environment. The problem is more realistic and is applicable in different industries where intermittent products cannot wait for further processing. Current work also considers heterogenous nature of machines where the machines are processing on different speeds rather than same which is practically hard to achieve. Numerical experiments are performed on small and large sized problem instances available on http://soa.iti.es. Tabu search is applied to 660 problem instances and results are compared with initial solution. It is because of the reason that as per our finding, no prior work is available in the literature which deals with this problem. Experiments are performed for heterogenous as well as for homogenous environment.

5.1 Introduction

One of the most widely studied area of operations research is scheduling (Potts et al, 2009). The paper published by Johnson (1954) is believed to be the very first one in scheduling which deals with two stage flowshop problem. It can be stated that history of flowshop is as old as that of scheduling. There are $(n!)^m$ possible solutions while scheduling jobs in a flowshop environment. In literature, order of jobs is kept same for all machines in the sequence and there is no job passing. This assumption reduces the number of possible schedules to only $n!$ because of which it is named as permutation flowshop. Permutation flowshop is based on assumptions as; 1) Processing time of each job on each machine is known. 2) Regardless of job's position in a sequence, set up times are included in processing time. 3) Only a single job can be processed on a machine at any given time. Finally, 4) Pre-emption of a job is not allowed on a machine. Gupta et.al (2006) reviewed the evolution of flowshop since very first publication by Johnson (1954) and discussed about solution approaches for the problem. For further study on permutation flowshop problem, review presented by Pan, Quan & Ruiz (2013) can be referred.

A generalization of permutation flowshop known as distributed permutation flowshop scheduling problem (DPFSP) is introduced by Naderi & Ruiz (2010). In DPFSP there are multiple parallel factories instead of one, as in conventional flowshop. Each factory consists of same number of machines and any job can be processed on either of the factory to perform all necessary operations. This demanding generalization was need of the time due to emergent market challenges and is being used in different industries (Gnoni et al. 2003; Leung et al. 2003; Sambasivan and Yahya 2005). Based on structure, distributed manufacturing facilities can be divided among two major categories, homogenous and heterogeneous. As names indicate, homogenous flowshops have similar machine in all aspects, operation and machine speed in all factories. In contrast, heterogeneous factories have set ups which differ from each other in either machine speed, labour skills etc.

There are some industries where jobs can't be delayed during processing between the machines due to nature of the products. These products include chemicals, metal, food and plastic. During manufacturing of these, intermittent material cannot be allowed to wait once processing is started on first machine, until it gets completed. This phenomena of not delaying or waiting is known as no-wait processing and in case of DPFSP it is called no-wait distributed permutation flowshop scheduling problem or no-wait DPFSP (NDPFSP). Following the three-field notation of scheduling problems, the stated problem for makespan objective can be designated as $DF_m|prmu$,

$nwt|C_{max}$. The NDPFSP will reduce to regular no-wait flow shop problem if all jobs are assigned to one factory (Lin & Ying, 2016). No-wait flowshop with more than two machines is an NP-hard problem (Röck, H, 1984) indicating that DPFSP with no-wait constraint is also an NP-hard problem.

Current work analyses the DPFSP with heterogeneous machine in different factories in terms of machine speed. Each machine of different factory has different speed. Due to realism, this paper also considers DPFSP with no-wait constraint beside heterogeneous nature of machines. So, a no-wait DPFSP or NDPFSP with heterogeneous machines is analyzed for makespan objective. A metaheuristic, tabu search (TS) is proposed to solve the problem.

Results of proposed tabu search are compared with existing metaheuristics. This chapter is organized in six sections. Section 5.2 introduces the problem description and formulation, while section 5.3 presents the solution methods. Experimental results for makespan objective are detailed in section 5.4. Finally, the chapter is concluded in section 5.5.

5.2 Problem description & formulation

The NDPFS problem's setting consists of F identical parallel factories or flow shops. Each factory consists of set of M machines where M= {1, 2,…, m). There are n number of jobs ($j = 1,2,3, \dots n$) with O operations to be performed on each job to make it a final product. The O operations for jobs are performed on M machines of a factory. Due to distributed no-wait flowshop nature of the problem, a job cannot be interrupted once it starts processing. A job j can start its processing on a machine $i \in M$ only in case if it has completed its operation on previous machine (i-1). Each machine (m^f) of every factory has different processing speed V_{if} where i=1, 2, 3, … m and f= 1,2 ,3 …. F. The processing time of a job changes from factory to factory and is inversely proportional to the machine speed. P_{ij} is the process time of job i at machine j if it is processed on standard machine which has speed 1. It takes p_{ij} time to perform j^{th} job at i^{th} machine of f^{th} factory where $p_{ij} = p_{ij}/ v_i$. A job can be assigned to any factory to perform all M operations. It is assumed that all operations are performed in the same factory. Thus, once assigned, a job is not allowed to transfer to any other factory. It is assumed that all parameters are deterministic. Objective involves determining the allocation of jobs to each factory and their sequence to achieve better results.

Since no-wait constraint is active between the factories, which requires to avoid any waiting time between consecutive operations. To satisfy this requirement, it's necessary to delay a job on the first machine before it starts processing so that all operations of a job are managed without any disruption. Computation of this delay time can be done using processing times of two consecutive jobs.

Following notations are used to represent the problem mathematically.

n Number of jobs

m Number of machines in each line

F Number of factories

K jobs position in the sequence $k \in \{1, 2, \dots, n\}$

i index of jobs $i \in \{1, 2, \dots, n\}$

f index of factories $f \in \{1, 2, \dots, F\}$

j index of machines $j \in \{1, 2, \dots, m\}$

P_{ij} processing time of job j at machine i

C_{max} Maximum completion time

Solution determines allocation of jobs and their order sequence σ^l for each factory f where σ is the sequence of jobs $\sigma = [\sigma^1, \sigma^2 ... \sigma^l]$. Expression for allocated jobs for factory 1 can be shown by $\sigma^1 = [\pi^1(1), \pi^1(2) \ldots \pi^l(n_1)]$ where n_1 is the number of jobs allocated to factory 1. The objective is to determine job sequence to minimize makespan. Speed variation of a machine can be accommodated by dividing the process time with speed of each machine at every factory.

$$P_{ij}^{f} = \frac{P_{ij}}{V_{jf}}$$

Where

V_{jf} = Speed of machine *j* of factory *f*

P_{ij}^{f} = Process time of job *i* at machine *j* of factory *f*

To avoid waiting of a job or introducing no-wait phenomenon, a job is required to wait before it starts processing for a certain calculated duration. This waiting duration is calculated based on the maximum difference of job and machine availability. Let $(\pi_1, \pi_2, \ldots \pi_n)$ denote the schedule for jobs in a factory then the start and finish time of first scheduled job is;

$$S(\pi_1, k) = C(\pi_1, k-1) \qquad \text{where} \qquad k=1, \ldots, m$$

$$C(\pi_1, k) = S(\pi_1, k) + P(\pi_1, k) \qquad \text{where} \qquad k=1, \ldots, m$$

Initialization

$$C(\pi_i, 0) = 0 \qquad \text{where} \qquad i=1, \ldots, n$$

The start time of other scheduled jobs $(\pi_2, \ldots \pi_n)$ for first machine are;

$$S(\pi_1, k) = max\{0, max_{k=1,\ldots,m}\ (C(\pi_{i-1}, k) \textstyle\sum_{j=1}^{k-1} P(\pi i, j))\} \qquad \text{where} \quad i=2, \ldots, n$$

Start time at other machines are;

$$S(\pi_i, k) = C(\pi_i, k-1) \qquad \text{where} \qquad k=2, \ldots, m$$

Completion time at all machines are;

$$C(\pi_i, k) = S(\pi_i, k) + P(\pi_i, k)$$

5.3 Solution method

We proposed a TS metaheuristic to solve the problem (originally proposed by Glover, 1986 to deal with combinatorial problems). Tabu search procedure applied for this problem is same as that of detailed in section 4.3 except that of added improvement scheme. Pseudocode of the TS is presented in Figure 13.

Generate an initial solution π using construction heuristic

Initialize best solution π* ← π
Initialize *InsertionCostMatrix* $A^{Net}(\pi)$
    Set Counter1=0, Counter2 =0
While termination criteria not met **do**
    Counter1 ++
    Counter2 ++
    Generate neighbourhood solutions *N(π)*
    Choose the best neighbourhood solution which is not tabu
    π'ϵN(π)
    Set π=π'
    Update *InsertionCostMatrix* $A^{Net}(\pi)$
    if Counter **=** β_1
        π ← IntraFactory LocalSearch
    Counter1 = 0
    end if
    if Counter **=** β_2
        π ← Complete LocalSearch
    Counter2 = 0
    end if
    if π is better than π*
        π* = π
    end if
end while
return π*

Figure 13 Pseudocode of tabu search

5.31 Initialize and updating insertion cost matrix

To reduce the CPU time of proposed TS, a database can help avoid repeated calculation of neighbourhood moves. An insertion cost matrix is proposed in this paper to fulfil this objective, as detailed in section 4.3. The matrix keeps record of increase or decrease in makespan of a factory if a certain job is added to or is removed from a factory. Three metrices known as A^+, A^- and A^{Net} are used to keep track of increase, decrease or net change in makespan of a factory due to movement of a certain job. Net cost matrix is updated based on change in makespan of a factory and is used further for next iterations. This procedure helps avoid repetition of the same procedure every time leading to reduction of CPU time.

5.3.2 Intra-factory and complete local search

Intra-factory and complete local searches are applied to find improved solution within a factory. This local search is called after every 25^{th} iteration. All jobs are randomly selected and are placed in all possible positions of a factory's schedule. After every iteration, previous solution is replaced with improved one and the process is repeated until solution keeps on improving. Also known as inter-factory local search, complete local search is performed to improve the solution quality by possible relocation of jobs in different factories.

5.4 Numerical analysis

In this section, performance of proposed algorithm, tabu search, is evaluated using computational experiments. This analysis is performed on a set of small & large sized data set presented by Naderi and Ruiz (2010) which are available in http://soa.iti.es. Small sized instances used are a set of {4, 6, 8, 10} jobs and {2, 3, 4, 5} machines. Number of factories is 3 for all problem instances. Each combination of jobs, machines and factories has 5 instances. Thus, a total of 80 instances were considered in small-sized instances. For large sized instances, analysis is performed on a set of 660 instances presented by Naderi and Ruiz (2010) which are also available in http://soa.iti.es. These instances are generated using a combination of number of jobs and machines are {20, 50, 100} $*$ 5, {20, 50, 100, 200} $*$ 10 and {20, 50, 100, 200} $*$ 20 and the number of factories F are $\{2, 3, 4, 5, 6, 7\}$. Processing times are based on uniform distribution of interval between [1, 99]. These processing times are divided by the speed of each machine to accommodate the heterogeneous nature of the problem. Tabu search is coded using C++ language to solve the problems. These problems are solved by running on a server with four 2.1 GHz processors having 16-core each and 256 GB of RAM memory. Tabu search is run for 5000 iterations with a tabu tenure λ to be 10. Based on computational time, stopping criteria is set to be *n. m. f.* 2 ms, indicating the stoppage of every algorithm when time is reached at this value.

5.4.1 Numerical analysis for small sized problems

This section provides results for eighty (80) small sized instances both for no-wait homogenous and heterogenous DPFSP problems. Makespan for these problems are compared with results of initial solution which are used as input for tabu search metaheuristics. These results are obtained by using equation below.

$$RPD_j^i = \left\{\left(H_j^i - I_i\right) / I_i \right\} * 100$$

where

RPD_j^i = Relative percentage deviation of the algorithm *j* from the best-known solution for problem instance *i*

H_j^i = Makespan obtained by algorithm *j* for problem instance *i*

I_i = Makespan obtained by the initial solution for problem instance *i*

Results of small sized instances both for homo and heterogeneous manufacturing environment are presented in Table 6. Initial solutions are created using construction heuristic. It can be observed from the table that tabu search improved 97.5% of instances and yield same results for only two instances which is 2.50% of total instances. For homogenous instances, it can be noticed that tabu search improved 64% of instances. At the same time, 20% of the instances are left unimproved. Results indicate that there is no relation in improvement between results of the same instances among both environments.

Table 6 Results for small instances

Instance	Heterogenous DPFSP Problems			Homogenous DPFSP Problems		
	Makespan		% Improvement	Makespan		% Improvement
	Initial	TS		Initial	TS	
1	553	495	11.72%	139	139	0.00%
2	652	591	10.32%	163	163	0.00%
3	664	606	9.57%	161	161	0.00%
4	724	622	16.40%	181	181	0.00%
5	600	544	10.29%	150	150	0.00%
6	704	647	8.81%	197	197	0.00%
7	691	670	3.13%	189	182	3.85%
8	820	820	0.00%	217	217	0.00%
9	781	627	24.56%	176	176	0.00%
10	743	706	5.24%	205	173	18.50%
11	1094	852	28.40%	293	263	11.41%
12	1133	1079	5.00%	304	291	4.47%

13	960	952	0.84%	302	300	0.67%
14	1176	1068	10.11%	301	291	3.44%
15	747	712	4.92%	268	235	14.04%
16	1294	1024	26.37%	390	390	0.00%
17	1114	937	18.89%	346	343	0.87%
18	1140	817	39.53%	327	327	0.00%
19	1019	892	14.24%	299	299	0.00%
20	1054	728	44.78%	278	278	0.00%
21	631	623	1.28%	166	161	3.11%
22	500	500	0.00%	125	125	0.00%
23	890	822	8.27%	252	214	17.76%
24	611	527	15.94%	145	137	5.84%
25	681	604	12.75%	181	164	10.37%
26	890	804	10.70%	226	222	1.80%
27	945	933	1.29%	257	256	0.39%
28	736	646	13.93%	185	185	0.00%
29	1015	923	9.97%	244	231	5.63%
30	1104	1046	5.54%	286	255	12.16%
31	1021	885	15.37%	275	251	9.56%
32	1094	929	17.76%	296	275	7.64%
33	1172	1045	12.15%	312	303	2.97%
34	1377	1236	11.41%	333	333	0.00%
35	1469	1223	20.11%	385	334	15.27%
36	1099	1003	9.57%	380	351	8.26%
37	928	858	8.16%	300	279	7.53%
38	1482	1239	19.61%	413	413	0.00%
39	1447	1194	21.19%	445	389	14.40%
40	853	625	36.48%	303	274	10.58%
41	883	840	5.12%	218	214	1.87%
42	780	702	11.11%	189	172	9.88%
43	688	658	4.56%	181	176	2.84%
44	985	767	28.42%	243	213	14.08%
45	997	777	28.31%	254	199	27.64%
46	1231	958	28.50%	304	272	11.76%
47	965	882	9.41%	256	241	6.22%
48	866	819	5.74%	253	234	8.12%

49	758	677	11.96%	237	196	20.92%
50	997	947	5.28%	260	255	1.96%
51	1311	1192	9.98%	356	343	3.79%
52	1322	1120	18.04%	316	313	0.96%
53	1115	1024	8.89%	303	287	5.57%
54	1338	1228	8.96%	380	342	11.11%
55	1214	1033	17.52%	326	312	4.49%
56	1312	1108	18.41%	382	355	7.61%
57	1086	957	13.48%	374	331	12.99%
58	1104	993	11.18%	371	348	6.61%
59	1324	1283	3.20%	424	410	3.41%
60	1382	1148	20.38%	427	376	13.56%
61	896	880	1.82%	219	210	4.29%
62	1042	846	23.17%	253	218	16.06%
63	894	848	5.42%	232	217	6.91%
64	1155	1006	14.81%	276	242	14.05%
65	876	827	5.93%	219	195	12.31%
66	1265	1070	18.22%	309	273	13.19%
67	1333	1269	5.04%	363	344	5.52%
68	1112	942	18.05%	308	257	19.84%
69	1109	949	16.86%	283	263	7.60%
70	1137	1018	11.69%	298	265	12.45%
71	1246	1104	12.86%	371	334	11.08%
72	1191	1125	5.87%	362	320	13.13%
73	1355	1197	13.20%	350	341	2.64%
74	1515	1345	12.64%	387	367	5.45%
75	1465	1338	9.49%	389	374	4.01%
76	1270	1062	19.59%	364	343	6.12%
77	1302	1144	13.81%	408	371	9.97%
78	1568	1328	18.07%	437	425	2.82%
79	1697	1360	24.78%	435	419	3.82%
80	1225	1183	3.55%	393	353	11.33%
Average	1048.99	925.10	13.35%	288.8125	270.35	6.86%

5.4.2 Numerical analysis for no-wait heterogenous DPFSP

Results for no-wait heterogenous DPFSP problem obtained with application of proposed tabu search are compared with initial solution for above mentioned 660 instances, using relative percentage deviation against initial solutions. Large-sized instances have 660 instances divided into eleven groups, based on a combination of number of jobs and number of machines. The average of relative percentage deviation is grouped by *n* x *m*. Each group has sixty (60) problem instances. The resulting RPD values based on mentioned stopping criteria (t=2) are presented in Table 7. It can be noticed that tabu search provided improved results both in terms of solution quality and CPU time. Average RPD value for initial solution is 8454.95 whereas for TS its 8173.33 resulting 4.59 % improvement on average. The TS provided improved solutions for 98% (652) instances and same results for 2.00% (8) instances.

Table 7 Results for no-wait heterogenous DPFSP

n x m	**Initial Solution**	**TS**	**RPD**
20 x 5	1947.05	1816.45	-7.62
20 x 10	2860.10	2647.13	-7.84
20 x 20	4771.53	4547.63	-4.84
50 x 5	3771.68	3598.75	-5.45
50 x 10	5163.58	4944.13	-4.77
50 x 20	7634.58	7355.97	-3.88
100 x 5	6840.57	6605.67	-3.98
100 x 10	9000.13	8719.55	-3.42
100 x 20	12432.17	12099.90	-2.95
200 x 10	16600.28	16107.37	-3.23
200 x 20	21982.80	21464.12	-2.53
Average	8454.952	8173.33	-4.59

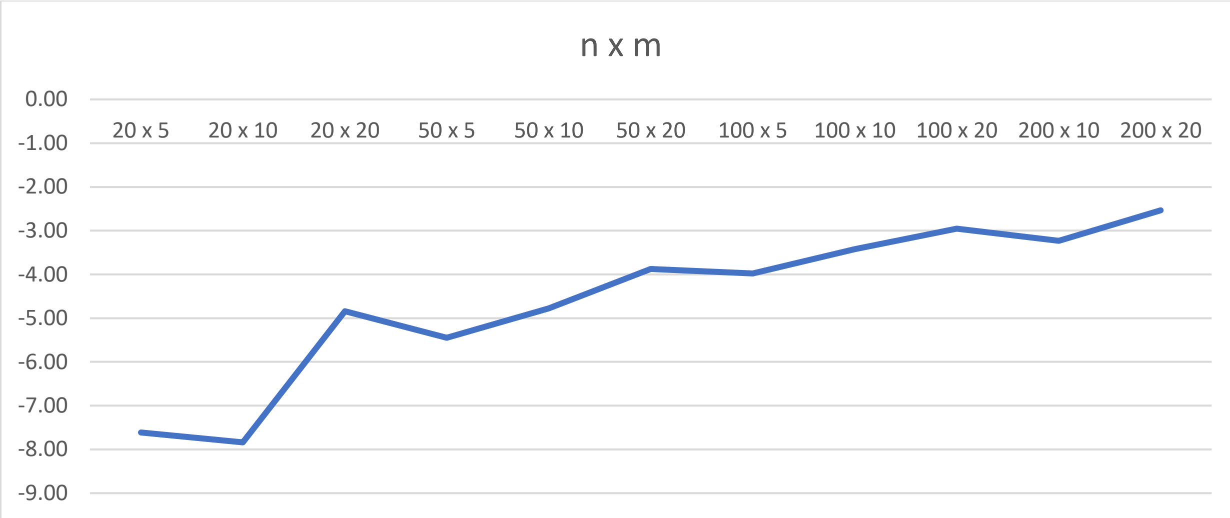

Figure 14 Relation between number jobs and relative percentage deviation

Figure 14 represents the relationship between relative percentage deviation and combination of *n x m*. It can be observed that RPD decreases with the increase in number of jobs and machines and vice versa.

5.4.3 Numerical analysis for no-wait homogenous DPFSP

Results for no-wait homogenous DPFSP problem of proposed tabu search are compared with General Variable Neighbourhood Search (GVNS) (Komaki & Malakooti, 2017) for above mentioned 660 instances using relative percentage deviation against best known solutions using below equation.

$$RPD_j^i = \{(H_j^i - B_i) \,/\, B_i\} * 100$$

where

RPD_j^i = Relative percentage deviation of the algorithm *j* from the best-known solution for problem instance *i*

H_j^i = Makespan obtained by algorithm *j* for problem instance *i*

B_i = Makespan obtained by the best-known solution provided by Komaki & Malakooti, (2017) for problem instance *i*

Results for no-wait homogenous DPFSP are presented in Table 8. This comparison is performed to evaluate the effectiveness of proposed tabu search. It can be noticed that with the increase in number of jobs, the RPD value increases. The RPD value is minimum for a combination of 100 jobs and 5 machines whereas its maximum for combination of 200 jobs.

Table 8 Results for no-wait homogenous DPFSP

n x m	**GVNS**	**TS**	**RPD**
20 x 5	0.83	545.00	3.74
20 x 10	2.35	848.05	3.25
20 x 20	2.64	1475.48	2.52
50 x 5	2.83	1031.87	3.15
50 x 10	3.71	1475.72	6.77
50 x 20	3.83	2254.58	3.86
100 x 5	1.40	1872.95	1.18
100 x 10	0.81	2524.63	1.81
100 x 20	0.22	3591.28	6.84
200 x 10	0.30	4597.32	13.46
200 x 20	0.20	6228.17	20.02
Average	1.74	2404.10	6.06

No relation is observed between number of jobs and relative percentage deviation of jobs.

5.5 Conclusion

Current research deals with distributed permutation flow shop scheduling problem for heterogeneous environments in terms of machine speed with additional constraint of no-wait. The DPFSP problem in general deals with solution of two problems, assignment of jobs to factories and sequencing of jobs for each factory to optimize the objective. No-wait is an additional constraint to be dealt besides solving these two problems of assignment and sequencing. A metaheuristic, tabu search is used to solve the problem. An insertion cost matrix is added to TS to evaluate the cost of any move of jobs within or between factories followed by an improvement scheme.

Results of small sized instances are compared with the results of initial solution. Since there is no benchmark results for heterogenous DPFSP problems, the results for large sized instances are compared with those of obtained by Komaki & Malakooti, (2017) for homogenous no-wait DPFSP problems. This comparison indicates that proposed method provides comparable results to those of provided by GVNS. In case of heterogeneous problems, none of the study is performed earlier for a problem with or without no-wait constraint. Results are compared with initial solution, obtained with application of construction heuristic. Tabu search provided an average of 4.59% improved results for 660 problem instances available on http://soa.iti.es. Since no results are available for the problem under consideration, the results achieved can be considered as benchmark. Current work can be extended in future for analysis of other variances of heterogeneous no-wait DPFSP including labour skills, factory eligibility constraint etc.

CHAPTER 6

Conclusion

Economic globalization and frequently changing market structures force organizations to move to geographically scattered, decentralized plants instead of centralized. Market competitiveness and demand for higher utilisation of resources can only be addressed by incorporating distributed manufacturing rather than centralized or single site in production fleets. By exploiting DPFSP environment, companies can reduce manufacturing costs, organisational risk and can improve the quality of products. Distributed manufacturing is a common methodology currently being used in different industries including automotive, apparel and the steel.

A mathematical model and tabu search is developed to solve the DPFSP problem for total completion time objective. The proposed TS uses two distinct features: a cost matrix and an improvement scheme. The cost matrix helps reduce the CPU time, while improvement scheme helps improve the solution quality. For tabu search, initial solution is generated by an iterative construction heuristic which uses best insertion cost procedure. Insertion cost matrix and improvement schemes are introduced to improve the results. Insertion cost matrix is a distinctive procedure which helps avoid repetitive calculation for repetitive solutions, leading to reduction in cost of metaheuristic. Proposed improvement scheme is based on three types of destroy and repair operators. Local search is applied to improve the resulting solution. Tabu search provided improved results for 57.22% instances as compared to existing metaheuristics.

Based on literature review, distributed permutation flow shop scheduling problem for heterogeneous environment in terms of machine speed with additional constraint of no-wait is also studied in this thesis. Results of small instances are compared with the results of initial solution. Since DPFSP for heterogenous problem is not studied before, results for large sized instances are compared with those of obtained by Komaki & Malakooti, (2017) for homogenous no-wait DPFSP problems using GVNS. Proposed method provides comparable results to those of provided by GVNS. For heterogeneous problems, the results are compared with initial solution, obtained with application of construction heuristic. Tabu search provided an average of 4.59% improved results. Since no results are available for the heterogenous no-wait DPFSP problem, the results achieved can be considered as benchmark.

One limitation of this thesis is that some of the practical constraints including labour skills, and factory eligibility are not considered. These constraints are common in DPFSP environment and can be considered in future studies to make the model more realistic.

References

1. Arabameri, S., & Salmasi, N. (2013). Minimization of weighted earliness and tardiness for no-wait sequence-dependent setup times flowshop scheduling problem. *Computers & Industrial Engineering*, *64*(4), 902-916.
2. Bagassi, S., Francia, D., & Persiani, C. A. (2010). Terminal area "green" optimization using job-shop scheduling techniques. In *Proceedings of the 27th congress of the international council of the aeronautical sciences 2010, ICAS 2010* (Vol. 6, pp. 5135-5143).
3. Bargaoui, H., Driss, O. B., & Ghédira, K. (2017). A novel chemical reaction optimization for the distributed permutation flowshop scheduling problem with makespan criterion. *Computers & Industrial Engineering*, *111*, 239-250.
4. Cai, S., Yang, K., & Liu, K. (2017). Multi-objective Optimization of the Distributed Permutation Flow Shop Scheduling Problem with Transportation and Eligibility Constraints. *Journal of the Operations Research Society of China*, 1-26.
5. Chan HK, Chung SH. 2013. Optimisation approaches for distributed scheduling problems. Int J Prod Res. 51(9):2571–2577.
6. Chan, F. T., Chung, S. H., & Chan, P. L. Y. (2005). An adaptive genetic algorithm with dominated genes for distributed scheduling problems. *Expert Systems with Applications*, *29*(2), 364-371.
7. Charles Stark Draper Laboratory. Automation Management Systems Division. (1984). *Flexible manufacturing systems handbook*. Park Ridge, N.J., USA: Noyes Publications.

8. Chau, V., Chen, X., Fong, K. C., Li, M., & Wang, K. (2017). Flow shop for dual CPUs in dynamic voltage scaling. *Theoretical Computer Science*.
9. De Giovanni, L., & Pezzella, F. (2010). An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. *European journal of operational research*, *200*(2), 395-408.
10. Deng, J., & Wang, L. (2017). A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem. *Swarm and Evolutionary Computation*, *32*, 121-131.
11. Deng, J., Wang, L., Wu, C., Wang, J., & Zheng, X. (2016, August). A Competitive Memetic Algorithm for Carbon-Efficient Scheduling of Distributed Flow-Shop. In *International Conference on Intelligent Computing* (pp. 476-488). Springer International Publishing.
12. Ding, J. Y., Song, S., & Wu, C. (2016). Carbon-efficient scheduling of flow shops by multi-objective optimization. *European Journal of Operational Research*, *248*(3), 758-771.
13. Dorigo, M. (1992). Optimization, Learning and Natural Algorithms. Ph.D. thesis, Politecnico di Milano, Italy.
14. Duan, W., Li, Z., Ji, M., Yang, Y., Wang, S., & Liu, B. (2016, July). A hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling with flowline eligibility. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 2581-2587). IEEE.
15. Fan, S. K. S. (2000). Quality improvement of chemical-mechanical wafer planarization process in semiconductor manufacturing using a combined generalized linear modelling-

non-linear programming approach. *International Journal of Production Research*, *38*(13), 3011-3029.

16. Fernandez-Viagas, V., & Framinan, J. M. (2015). A bounded-search iterated greedy algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Production Research*, *53*(4), 1111-1123.
17. Fernandez-Viagas, V., Perez-Gonzalez, P., & Framinan, J. M. (2018). The distributed permutation flow shop to minimise the total flowtime. *Computers & Industrial Engineering*, *118*, 464-477.
18. Gajpal, Y., & Rajendran, C. (2006). An ant-colony optimization algorithm for minimizing the completion-time variance of jobs in flowshops. *International journal of production economics*, *101*(2), 259-272.
19. Gajpal, Y., Rajendran, C., & Ziegler, H. (2006). An ant colony algorithm for scheduling in flowshops with sequence-dependent setup times of jobs. *The international journal of advanced manufacturing technology*, *30*(5-6), 416-424.
20. Gao, J., & Chen, R. (2011). A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Computational Intelligence Systems*, *4*(4), 497-508.
21. Gao, J., Chen, R., & Deng, W. (2013). An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Production Research*, *51*(3), 641-651.
22. Gao, J., Chen, R., Deng, W., & Liu, Y. (2012). Solving multi-factory flowshop problems with a novel variable neighbourhood descent algorithm. *Journal of Computational Information Systems*, *8*(5), 2025-2032.

23. Glover F. 1993. Future paths for integer programming and links to artificial intelligence. Comput Oper Res. 13(5):533–549.

24. Gnoni MG, Iavagnilio R, Mossa G, Mummolo G, Di Leva A. 2003. Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry. Int J Prod Econ. 85(2):251–262.

25. Goldman, S. L., Preiss, K., Nagel, R., & Dove, R. (1991). 21st century manufacturing enterprise strategy: An industry-led view. *Iacocca Institute, Lehigh University, Bethlehem, PA*, *106*.

26. Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations research*, *44*(3), 510-525.

27. Hatami, S. (2016). *The Distributed and Assembly Scheduling Problem* (Doctoral dissertation).

28. Hecker, F. T., Stanke, M., Becker, T., & Hitzmann, B. (2014). Application of a modified GA, ACO and a random search procedure to solve the production scheduling of a case study bakery. *Expert Systems with Applications*, *41*(13), 5882-5891.

29. Hsu, V. N., De Matta, R., & Lee, C. Y. (2003). Scheduling patients in an ambulatory surgical center. *Naval Research Logistics (NRL)*, *50*(3), 218-238.

30. Jia, H. Z., Fuh, J. Y., Nee, A. Y., & Zhang, Y. F. (2002). Web-based multi-functional scheduling system for a distributed manufacturing environment. *Concurrent Engineering*, *10*(1), 27-39.

31. Jia, H. Z., Fuh, J. Y., Nee, A. Y., & Zhang, Y. F. (2007). Integration of genetic algorithm and Gantt chart for job shop scheduling in distributed manufacturing systems. *Computers & Industrial Engineering*, *53*(2), 313-320.

32. Jia, H. Z., Nee, A. Y., Fuh, J. Y., & Zhang, Y. F. (2003). A modified genetic algorithm for distributed scheduling problems. *Journal of Intelligent Manufacturing*, *14*(3-4), 351-362.

33. Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly*, *1*(1), 61-68.

34. Kahn KB. 2012. The PDMA handbook of new product development: John Wiley and Sons.

35. Kahn, K. B., Castellion, G. A., and Griffin, A. (2004). The PDMA handbook of new product development. Wiley, New York, Second edition.

36. Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. IEEE International Conference on Neural Networks, 4:1942–1948.

37. Khedr, A. M., & Osamy, W. (2011). Minimum perimeter coverage of query regions in a heterogeneous wireless sensor network. *Information Sciences*, *181*(15), 3130-3142.

38. Kim, J., Kröller, A., & Mitchell, J. (2009). Scheduling aircraft to reduce controller workload. In *9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'09)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

39. Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. Science, 220(4598):671–680.

40. Koestler, A. (1968). The ghost in the machine.

41. KołlOdziej, J., & Khan, S. U. (2012). Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment. *Information Sciences*, *214*, 1-19.

42. Komaki, G. M., Mobin, S., Teymourian, E., & Sheikh, S. (2015). A General Variable Neighborhood Search Algorithm to Minimize Makespan of the Distributed Permutation Flowshop Scheduling Problem. *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*,*9*(8), 2618-2625.

43. Komaki, M., & Malakooti, B. (2017). General variable neighborhood search algorithm to minimize makespan of the distributed no-wait flow shop scheduling problem. *Production Engineering*, 1-15.

44. Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., & Van Brussel, H. (1999). Reconfigurable manufacturing systems. *CIRP Annals-Manufacturing Technology*, *48*(2), 527-540.

45. Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems). The MIT Press.

46. Kühnle, H. (Ed.). (2009). *Distributed manufacturing: paradigm, concepts, solutions and examples*. Springer Science & Business Media.

47. Leung SCH, Wu Y, Lai KK. 2003. Multi-site aggregate production planning with multiple objectives: A goal programming approach. Prod Plan Control. 14(5):425–436.

48. Li, H., Gajpal, Y., & Bector, C. R. (2018). Single machine scheduling with two-agent for total weighted completion time objectives. *Applied Soft Computing*, *70*, 147-156.

49. Li Y, Chen Z. 2015. The distributed permutation flowshop scheduling problem: A genetic algorithm approach. (Icmii):381–384.

50. Li, Y., & Chen, Z. (2015). The distributed permutation flowshop scheduling problem: A genetic algorithm approach. *3rd International Conference on Mechatronics and Industrial Informatics*, 381-384.

51. Li, Z., Duan, W., Ji, M., Yang, Y., Wang, S., & Liu, B. (2016, July). The distributed permutation flowshop scheduling problem with different transport timetables and loading capacities. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 2433-2437). IEEE.

52. Lin SW, Ying KC, Huang CY. 2013. Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm. Int J Prod Res. 51(16):5029–5038.

53. Lin, S. W., & Ying, K. C. (2016). Minimizing makespan for solving the distributed no-wait flowshop scheduling problem. *Computers & Industrial Engineering*, *99*, 202-209.

54. Lin, S. W., Lee, Z. J., Ying, K. C., and Lu, C. C. (2011). Minimization of maximum lateness on parallel machines with sequence-dependent setup times and job release dates. Computers & Operations Research, 38(5):809–815.

55. Liu H, Gao L. 2010. A discrete electromagnetism-like mechanism algorithm for solving distributed permutation flowshop scheduling problem. Proc - 2010 Int Conf Manuf Autom ICMA 2010.:156–163.

56. Liu, H., & Gao, L. (2010). A discrete electromagnetism-like mechanism algorithm for solving distributed permutation flowshop scheduling problem. In *Manufacturing Automation (ICMA), International Conference* (pp. 156-163). IEEE.

57. Liu, S. Q., & Kozan, E. (2011). Scheduling trains with priorities: a no-wait blocking parallel-machine job-shop scheduling model. *Transportation Science*, *45*(2), 175-198.

58. Mannino, C., & Mascis, A. (2009). Optimal real-time traffic control in metro stations. *Operations Research*, *57*(4), 1026-1039.

59. Naderi, B., & Ruiz, R. (2010). The distributed permutation flowshop scheduling problem. *Computers & Operations Research*, *37*(4), 754-768.

60. Naderi, B., & Ruiz, R. (2014). A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European Journal of Operational Research*, *239*(2), 323-334.

61. Osman, I. H., & Potts, C. N. (1989). Simulated annealing for permutation flow-shop scheduling. *Omega*, *17*(6), 551-557.

62. Peklenik, J. (1992). FMS-A complex object of control. In *Proceedings of the Eighth International Prolomat Conference*(pp. 1-25).

63. Pinedo, M. (2012). *Scheduling* (Vol. 29). New York: Springer.

64. Reeves, C. R. (1993). *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc.

65. Ropke S, Pisinger D. 2006. A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. Eur J Oper Res. 171(3):750–775.

66. Ruiz, R., Pan, Q. K., & Naderi, B. (2019). Iterated Greedy methods for the distributed permutation flowshop scheduling problem. *Omega*, *83*, 213-222.

67. Sahu, S. N., Gajpal, Y., & Debbarma, S. (2018). Two-agent-based single-machine scheduling with switchover time to minimize total weighted completion time and makespan objectives. *Annals of Operations Research*, *269*(1-2), 623-640.

68. Sambasivan M, Yahya S. 2005. A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. Comput Oper Res. 32(3):537–555.

69. Shabtay, D., Bensoussan, Y., & Kaspi, M. (2012). A bicriteria approach to maximize the weighted number of just-in-time jobs and to minimize the total resource consumption cost in a two-machine flow-shop scheduling system. *International Journal of Production Economics*, *136*(1), 67-74.

70. Shao, W., Pi, D., & Shao, Z. (2017). Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms. *Knowledge-Based Systems*, *137*, 163-181.

71. Ueda, K. (1992). A concept for bionic manufacturing systems based on DNA-type information. In *Human Aspects in Computer Integrated Manufacturing* (pp. 853-863).

72. Wang FY, Chua TJ, Cai TX, Chai LS. 2007. Common capacity modelling for multi-site planning: Case studies. IEEE Int Conf Emerg Technol Fact Autom ETFA.:336–343.

73. Wang J, Wang L, Shen J. 2016. A hybrid discrete cuckoo search for distributed permutation flowshop scheduling problem. 2016 IEEE Congr Evol Comput CEC. (2013):2240–2246.

74. Wang SY, Wang L, Liu M, Xu Y. 2013. An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. Int J Prod Econ. 145(1):387–396.

75. Wang, B., Han, X., Zhang, X., & Zhang, S. (2015). Predictive-reactive scheduling for single surgical suite subject to random emergency surgery. *Journal of Combinatorial Optimization*, *30*(4), 949-966.

76. Wang, J. J., & Wang, L. (2018). A Knowledge-Based Cooperative Algorithm for Energy-Efficient Scheduling of Distributed Flow-Shop. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

77. Wang, J., Wang, L., & Shen, J. (2016, July). A hybrid discrete cuckoo search for distributed permutation flowshop scheduling problem. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 2240-2246). IEEE.

78. Wang, K., Huang, Y., & Qin, H. (2016). A fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling problems under machine breakdown. *Journal of the Operational Research Society*, *67*(1), 68-82.

79. Wang, S. Y., Wang, L., Liu, M., & Xu, Y. (2013). An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *International Journal of Production Economics*, *145*(1), 387-396.

80. Wang, Y., Tang, J., Pan, Z., & Yan, C. (2015). Particle swarm optimization-based planning and scheduling for a laminar-flow operating room with downstream resources. *Soft Computing*, *19*(10), 2913-2926.

81. Warnecke, H. J. (2003). Fractal Company—A Revolution in Corporate Culture. In *Manufacturing Technologies for Machines of the Future* (pp. 63-85). Springer, Berlin, Heidelberg.

82. Wen, Y., Xu, H., & Yang, J. (2011). A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system. *Information Sciences*, *181*(3), 567-581.

83. Wilkinson SJ, Cortier A, Shah N, Pantelides CC. 1996. Integrated production and distribution scheduling on a Europe-wide basis. Comput Chem Eng. 20(96):S1275–S1280.

84. Xu Y, Wang L, Wang S, Liu M. 2014. An effective hybrid immune algorithm for solving the distributed permutation flow-shop scheduling problem. Eng Optim. 46(9):1269–1283.

85. Xu, Y., Wang, L., Wang, S., & Liu, M. (2014). An effective hybrid immune algorithm for solving the distributed permutation flow-shop scheduling problem. *Engineering Optimization*, *46*(9), 1269-1283.

Appendix A: Results of DPFSP problems in chapter 4

Problems				Results		
No	f	n	m	TCT	RPD	CPU (S)
1	2	20	5	9195	0.00	0.67
2	2	20	5	9998	-0.04	0.67
3	2	20	5	8733	0.00	0.67
4	2	20	5	10305	0.00	0.67
5	2	20	5	8955	0.00	0.67
6	2	20	5	8708	-0.02	0.67
7	2	20	5	8809	-0.12	0.67
8	2	20	5	9219	0.00	0.67
9	2	20	5	9592	0.00	0.67
10	2	20	5	8561	-0.22	0.67
11	2	20	10	15285	0.00	1.67
12	2	20	10	16594	-0.04	1.67
13	2	20	10	14553	0.00	1.33
14	2	20	10	13635	0.00	1.67
15	2	20	10	13673	0.00	1.33
16	2	20	10	13909	-0.14	1.33
17	2	20	10	13320	0.00	1.67
18	2	20	10	14776	0.00	1.33
19	2	20	10	14711	-0.04	1.33
20	2	20	10	15497	-0.03	1.33
21	2	20	20	26807	0.00	2.67
22	2	20	20	25195	0.00	3.00
23	2	20	20	27036	-0.01	3.00
24	2	20	20	25294	0.00	2.67
25	2	20	20	27324	0.00	2.67
26	2	20	20	25956	0.00	3.00
27	2	20	20	26357	0.00	3.00
28	2	20	20	25660	0.00	3.00
29	2	20	20	26874	0.00	3.00
30	2	20	20	25427	0.00	3.00
31	2	50	5	38013	-0.06	10.33
32	2	50	5	39836	-0.44	11.00
33	2	50	5	37072	-0.50	10.33
34	2	50	5	40027	-0.77	11.00
35	2	50	5	40827	-0.32	10.33
36	2	50	5	39398	0.17	11.00

Problems				Results		
No	f	n	m	TCT	RPD	CPU (S)
361	5	20	5	6398	0.00	0.33
362	5	20	5	6701	0.00	0.00
363	5	20	5	5908	0.00	0.00
364	5	20	5	7113	-0.02	0.33
365	5	20	5	6179	0.00	0.33
366	5	20	5	6074	0.00	0.33
367	5	20	5	6040	0.00	0.33
368	5	20	5	6387	0.00	0.33
369	5	20	5	6625	0.00	0.33
370	5	20	5	5897	0.00	0.33
371	5	20	10	11658	0.00	0.33
372	5	20	10	12677	0.00	0.33
373	5	20	10	11251	0.00	0.33
374	5	20	10	10473	0.00	0.33
375	5	20	10	10689	0.00	0.67
376	5	20	10	10584	0.00	0.33
377	5	20	10	10491	0.00	0.67
378	5	20	10	11313	0.00	0.33
379	5	20	10	11253	0.00	0.33
380	5	20	10	11991	0.00	0.33
381	5	20	20	22190	0.00	1.00
382	5	20	20	20859	0.00	1.00
383	5	20	20	22381	0.00	1.00
384	5	20	20	21162	0.00	1.00
385	5	20	20	22563	0.00	1.00
386	5	20	20	21513	0.00	1.00
387	5	20	20	21988	0.00	1.00
388	5	20	20	21384	0.00	1.00
389	5	20	20	22096	0.00	1.00
390	5	20	20	21165	0.00	1.00
391	5	50	5	21373	-0.18	3.67
392	5	50	5	22518	-0.09	3.67
393	5	50	5	21052	-0.09	3.67
394	5	50	5	22874	0.08	3.67
395	5	50	5	22943	-0.11	3.67
396	5	50	5	22238	0.00	3.67

37	2	50	5	39012	-0.33	10.67
38	2	50	5	37923	0.15	11.00
39	2	50	5	37147	-0.16	10.67
40	2	50	5	40513	-0.21	10.33
41	2	50	10	56888	0.37	23.67
42	2	50	10	53825	-0.41	24.00
43	2	50	10	51910	-0.57	23.33
44	2	50	10	55886	-0.39	22.33
45	2	50	10	56249	-0.58	23.67
46	2	50	10	56290	-0.53	24.00
47	2	50	10	57622	-0.32	22.00
48	2	50	10	56467	0.17	23.33
49	2	50	10	55866	0.09	22.67
50	2	50	10	57203	-0.60	23.67
51	2	50	20	90547	0.17	48.33
52	2	50	20	84970	-0.26	47.33
53	2	50	20	83951	-0.13	46.33
54	2	50	20	86408	-0.21	48.00
55	2	50	20	85196	-0.28	45.67
56	2	50	20	86564	0.21	43.67
57	2	50	20	88053	0.70	46.33
58	2	50	20	87807	-0.13	45.33
59	2	50	20	87525	-0.09	45.33
60	2	50	20	88673	-0.25	44.67
61	2	100	5	141099	-0.40	100.67
62	2	100	5	135410	-0.12	94.67
63	2	100	5	131983	-0.41	94.33
64	2	100	5	126445	-0.46	101.33
65	2	100	5	133192	-0.44	98.00
66	2	100	5	129645	-0.27	107.33
67	2	100	5	132367	-0.96	101.00
68	2	100	5	128578	-0.63	93.33
69	2	100	5	138514	-0.43	96.33
70	2	100	5	135241	-0.42	103.33
71	2	100	10	178872	-0.29	214.00
72	2	100	10	163482	-0.61	198.00
73	2	100	10	173353	-0.42	201.00
74	2	100	10	181484	-0.31	197.33
75	2	100	10	171147	-0.19	192.67

397	5	50	5	22077	-0.09	3.33
398	5	50	5	21625	-0.04	3.67
399	5	50	5	20983	-0.09	3.33
400	5	50	5	22834	0.04	3.33
401	5	50	10	36890	-0.03	7.33
402	5	50	10	35200	0.14	7.00
403	5	50	10	34161	-0.10	7.33
404	5	50	10	36640	-0.07	7.33
405	5	50	10	36798	0.00	7.00
406	5	50	10	36605	-0.10	7.33
407	5	50	10	37390	-0.11	6.67
408	5	50	10	36578	0.04	7.00
409	5	50	10	36534	0.04	7.33
410	5	50	10	37675	0.03	7.00
411	5	50	20	66283	-0.09	13.00
412	5	50	20	62403	0.03	13.67
413	5	50	20	61470	0.01	13.67
414	5	50	20	63472	0.42	13.00
415	5	50	20	62469	0.00	12.67
416	5	50	20	63070	-0.01	13.00
417	5	50	20	64299	-0.02	13.67
418	5	50	20	64240	0.25	13.00
419	5	50	20	63928	-0.05	13.00
420	5	50	20	65007	0.05	13.00
421	5	100	5	70390	0.24	32.00
422	5	100	5	67215	-0.12	31.33
423	5	100	5	65474	-0.20	28.67
424	5	100	5	63559	0.05	29.00
425	5	100	5	66542	0.04	28.67
426	5	100	5	64862	0.26	32.67
427	5	100	5	66077	-0.09	30.00
428	5	100	5	63917	-0.42	29.33
429	5	100	5	69401	0.25	31.67
430	5	100	5	67593	0.19	31.00
431	5	100	10	102051	0.00	63.67
432	5	100	10	93133	-0.17	65.33
433	5	100	10	98921	-0.02	54.33
434	5	100	10	103113	-0.06	61.00
435	5	100	10	98143	-0.06	58.33

76	2	100	10	163495	0.30	204.33
77	2	100	10	167885	-0.39	198.33
78	2	100	10	173937	-0.74	202.33
79	2	100	10	180937	-0.40	202.33
80	2	100	10	175153	-0.52	203.33
81	2	100	20	242254	-0.31	442.33
82	2	100	20	245699	-0.30	429.33
83	2	100	20	243572	-0.58	435.00
84	2	100	20	245598	-0.63	440.67
85	2	100	20	243830	0.06	457.33
86	2	100	20	244611	-0.46	442.67
87	2	100	20	246641	-0.39	423.00
88	2	100	20	251947	-0.84	418.00
89	2	100	20	247624	0.08	422.00
90	2	100	20	249345	-0.30	410.67
91	2	200	10	590411	-0.57	2127.33
92	2	200	10	583297	-0.78	2079.67
93	2	200	10	587351	-1.06	2129.00
94	2	200	10	577686	-0.72	2091.33
95	2	200	10	582575	-0.80	2122.67
96	2	200	10	568477	-0.87	2133.67
97	2	200	10	600090	-0.24	2050.67
98	2	200	10	586716	-1.07	2180.33
99	2	200	10	576271	-1.07	2114.00
100	2	200	10	580961	-0.97	2078.00
101	2	200	20	744428	-0.49	4054.67
102	2	200	20	753708	-0.85	4339.33
103	2	200	20	765541	-1.18	4353.67
104	2	200	20	750842	-0.77	4363.67
105	2	200	20	743088	-0.59	4500.00
106	2	200	20	743815	-0.51	4337.33
107	2	200	20	755110	-0.57	4352.33
108	2	200	20	754850	-0.70	4100.00
109	2	200	20	747839	-0.94	4336.67
110	2	200	20	754979	-0.95	4549.67
111	2	500	20	3732769	-1.12	104991.33
112	2	500	20	3788562	-1.52	104253.00
113	2	500	20	3762837	-1.03	106126.33
114	2	500	20	3756351	-1.60	103038.67

436	5	100	10	92712	-0.13	62.33
437	5	100	10	95558	-0.12	56.67
438	5	100	10	98916	-0.16	63.33
439	5	100	10	102637	-0.03	60.67
440	5	100	10	100196	-0.44	62.00
441	5	100	20	156429	-0.54	122.00
442	5	100	20	160371	0.27	123.00
443	5	100	20	158320	-0.15	121.67
444	5	100	20	158865	-0.15	119.33
445	5	100	20	157007	-0.10	119.67
446	5	100	20	158282	-0.03	120.33
447	5	100	20	158505	-0.11	112.67
448	5	100	20	164973	0.19	124.00
449	5	100	20	159737	-0.24	121.00
450	5	100	20	161397	0.06	118.67
451	5	200	10	299396	0.24	653.67
452	5	200	10	295731	-0.01	650.00
453	5	200	10	298940	0.12	690.00
454	5	200	10	291701	-0.08	655.33
455	5	200	10	294485	-0.50	644.67
456	5	200	10	288983	-0.24	676.00
457	5	200	10	303034	-0.31	647.00
458	5	200	10	297965	0.02	651.33
459	5	200	10	290699	-0.64	643.67
460	5	200	10	296596	0.10	670.33
461	5	200	20	426224	0.04	1356.33
462	5	200	20	435283	0.22	1274.33
463	5	200	20	440879	-0.07	1295.67
464	5	200	20	430096	-0.04	1288.00
465	5	200	20	423431	-0.47	1297.33
466	5	200	20	426136	-0.21	1337.33
467	5	200	20	433150	-0.09	1323.00
468	5	200	20	433405	-0.19	1318.33
469	5	200	20	428554	-0.66	1317.00
470	5	200	20	433949	-0.25	1350.33
471	5	500	20	1862691	-0.52	45804.33
472	5	500	20	1895802	-0.83	38807.67
473	5	500	20	1875254	-0.62	38793.00
474	5	500	20	1882309	-0.76	39287.67

115	2	500	20	3765969	-0.79	105209.33
116	2	500	20	3754568	-1.07	103814.00
117	2	500	20	3729653	-1.28	104824.33
118	2	500	20	3777439	-1.01	104449.33
119	2	500	20	3727885	-1.29	107205.33
120	2	500	20	3769829	-1.00	106254.00
121	3	20	5	7650	-0.01	0.67
122	3	20	5	8226	0.02	0.33
123	3	20	5	7217	-0.04	0.33
124	3	20	5	8531	0.00	0.33
125	3	20	5	7398	0.00	0.33
126	3	20	5	7184	0.00	0.33
127	3	20	5	7303	-0.06	0.33
128	3	20	5	7644	-0.01	0.33
129	3	20	5	7966	0.00	0.33
130	3	20	5	7117	0.14	0.67
131	3	20	10	13329	0.00	1.00
132	3	20	10	14507	-0.03	1.00
133	3	20	10	12758	0.00	1.00
134	3	20	10	11937	0.00	1.00
135	3	20	10	12119	0.00	1.00
136	3	20	10	12032	-0.02	1.00
137	3	20	10	11706	0.00	1.00
138	3	20	10	12889	0.00	1.00
139	3	20	10	12796	-0.01	1.00
140	3	20	10	13564	0.00	1.00
141	3	20	20	24247	0.00	1.67
142	3	20	20	22839	0.00	1.67
143	3	20	20	24478	0.00	1.67
144	3	20	20	23060	0.00	1.67
145	3	20	20	24764	0.00	2.00
146	3	20	20	23543	0.00	1.67
147	3	20	20	24035	0.00	2.00
148	3	20	20	23292	0.00	2.00
149	3	20	20	24269	-0.07	1.67
150	3	20	20	23072	0.00	1.67
151	3	50	5	28875	0.18	9.00
152	3	50	5	30217	0.02	9.33
153	3	50	5	28247	-0.03	7.33
154	3	50	5	30522	-0.33	7.00
155	3	50	5	30988	-0.33	7.33

475	5	500	20	1869439	-0.82	39681.67
476	5	500	20	1871958	-0.66	39325.33
477	5	500	20	1861665	-0.56	36902.33
478	5	500	20	1879403	-0.90	37892.67
479	5	500	20	1867126	-0.50	38853.00
480	5	500	20	1873972	-0.74	39168.67
481	6	20	5	6089	0.00	0.33
482	6	20	5	6350	0.00	0.33
483	6	20	5	5591	0.02	0.33
484	6	20	5	6745	0.00	0.33
485	6	20	5	5885	0.03	0.33
486	6	20	5	5800	0.00	0.33
487	6	20	5	5738	0.03	0.33
488	6	20	5	6101	0.00	0.33
489	6	20	5	6278	0.00	0.33
490	6	20	5	5604	0.00	0.33
491	6	20	10	11305	0.00	1.00
492	6	20	10	12236	0.00	1.00
493	6	20	10	10904	0.00	1.00
494	6	20	10	10109	0.00	0.67
495	6	20	10	10306	0.00	1.00
496	6	20	10	10217	0.00	0.67
497	6	20	10	10218	0.00	0.67
498	6	20	10	10955	0.00	1.00
499	6	20	10	10906	0.00	1.00
500	6	20	10	11616	0.00	0.67
501	6	20	20	21712	0.00	1.67
502	6	20	20	20347	0.00	1.67
503	6	20	20	21896	0.00	1.67
504	6	20	20	20698	0.00	1.67
505	6	20	20	22063	0.00	1.67
506	6	20	20	21021	0.00	1.67
507	6	20	20	21459	0.00	1.33
508	6	20	20	20888	0.00	1.67
509	6	20	20	21600	0.00	1.67
510	6	20	20	20691	0.00	1.33
511	6	50	5	19567	0.16	3.00
512	6	50	5	20582	-0.40	5.00
513	6	50	5	19272	-0.04	5.33
514	6	50	5	20910	0.22	5.33
515	6	50	5	20909	-0.27	5.00

156	3	50	5	29956	0.66	7.33
157	3	50	5	29634	-0.40	6.67
158	3	50	5	28903	0.17	10.33
159	3	50	5	28126	-0.41	6.67
160	3	50	5	30752	0.05	6.00
161	3	50	10	45680	0.01	14.00
162	3	50	10	43883	0.60	15.67
163	3	50	10	42197	0.02	14.67
164	3	50	10	45405	-0.10	13.67
165	3	50	10	45682	0.00	14.00
166	3	50	10	45598	-0.26	13.67
167	3	50	10	46653	-0.13	13.00
168	3	50	10	45295	-0.36	14.67
169	3	50	10	45402	0.39	17.00
170	3	50	10	46728	0.13	13.67
171	3	50	20	77272	0.15	30.33
172	3	50	20	72768	-0.07	25.33
173	3	50	20	71820	-0.23	28.67
174	3	50	20	73758	-0.39	27.33
175	3	50	20	73004	0.17	27.00
176	3	50	20	73584	-0.03	26.00
177	3	50	20	74722	-0.09	28.33
178	3	50	20	74804	-0.43	28.33
179	3	50	20	74602	-0.37	25.67
180	3	50	20	75891	0.03	24.67
181	3	100	5	102445	0.50	72.67
182	3	100	5	97396	-0.28	61.00
183	3	100	5	95704	0.31	54.33
184	3	100	5	91683	-0.27	58.33
185	3	100	5	96317	-0.13	59.00
186	3	100	5	93482	-0.33	70.00
187	3	100	5	95694	-0.36	60.33
188	3	100	5	93235	0.13	56.67
189	3	100	5	100049	-0.21	65.67
190	3	100	5	97596	-0.21	58.33
191	3	100	10	137543	0.40	134.33
192	3	100	10	125125	-0.26	145.33
193	3	100	10	132664	-0.14	129.67
194	3	100	10	138522	0.01	139.00
195	3	100	10	130101	-0.93	123.67
196	3	100	10	123455	-0.81	146.33

516	6	50	5	20378	-0.05	5.67
517	6	50	5	20211	-0.04	5.33
518	6	50	5	19796	0.08	5.00
519	6	50	5	19172	-0.11	5.00
520	6	50	5	20839	-0.04	5.00
521	6	50	10	34512	-0.29	10.00
522	6	50	10	32936	-0.13	9.67
523	6	50	10	32228	0.02	10.33
524	6	50	10	34448	0.06	10.00
525	6	50	10	34577	0.08	9.67
526	6	50	10	34407	-0.05	10.00
527	6	50	10	35040	-0.07	9.67
528	6	50	10	34284	-0.14	10.33
529	6	50	10	34221	-0.19	10.00
530	6	50	10	35348	0.07	10.00
531	6	50	20	63520	-0.14	19.00
532	6	50	20	59698	0.00	18.67
533	6	50	20	58693	-0.08	19.67
534	6	50	20	60409	0.07	19.00
535	6	50	20	59773	0.09	18.00
536	6	50	20	60330	0.00	18.00
537	6	50	20	61612	0.09	19.00
538	6	50	20	61344	0.07	15.67
539	6	50	20	61101	0.06	19.67
540	6	50	20	62213	0.25	16.67
541	6	100	5	62399	0.03	45.33
542	6	100	5	59749	0.03	44.67
543	6	100	5	58329	0.23	44.33
544	6	100	5	56315	-0.27	39.67
545	6	100	5	59279	0.44	41.33
546	6	100	5	57433	0.01	35.00
547	6	100	5	58525	0.00	41.67
548	6	100	5	56648	-0.39	42.33
549	6	100	5	61533	0.11	46.33
550	6	100	5	59926	0.12	33.00
551	6	100	10	93117	0.17	89.33
552	6	100	10	84932	-0.29	53.00
553	6	100	10	90372	-0.05	80.00
554	6	100	10	93868	-0.16	81.00
555	6	100	10	89343	-0.31	51.33
556	6	100	10	84734	-0.19	56.67

197	3	100	10	127849	-0.42	117.67
198	3	100	10	133269	-0.11	133.00
199	3	100	10	137679	-0.40	134.67
200	3	100	10	134955	0.11	138.67
201	3	100	20	196400	0.24	261.33
202	3	100	20	199355	0.16	254.67
203	3	100	20	197207	-0.33	266.67
204	3	100	20	199523	0.04	259.67
205	3	100	20	196020	-0.22	270.00
206	3	100	20	197551	-0.25	271.67
207	3	100	20	199011	-0.16	247.67
208	3	100	20	205245	-0.24	276.00
209	3	100	20	199437	-0.30	271.33
210	3	100	20	202108	0.08	264.33
211	3	200	10	427897	-0.52	1490.00
212	3	200	10	424837	-0.36	1336.33
213	3	200	10	428097	-0.37	1418.33
214	3	200	10	420270	-0.17	1325.67
215	3	200	10	423434	-0.47	1392.33
216	3	200	10	415659	-0.04	1365.33
217	3	200	10	433848	-0.69	1403.33
218	3	200	10	427030	-0.65	1349.67
219	3	200	10	418742	-0.64	1367.00
220	3	200	10	423827	-0.44	1368.00
221	3	200	20	570585	-0.22	2895.00
222	3	200	20	578360	-0.69	2807.67
223	3	200	20	587657	-0.56	2906.67
224	3	200	20	575477	-0.55	2758.67
225	3	200	20	566987	-0.74	2876.33
226	3	200	20	567672	-0.85	2918.67
227	3	200	20	579037	-0.31	2650.00
228	3	200	20	578780	-0.38	2765.00
229	3	200	20	573493	-0.56	2872.67
230	3	200	20	579780	-0.48	2848.33
231	3	500	20	2702845	-0.99	65390.00
232	3	500	20	2749663	-1.19	66391.00
233	3	500	20	2726796	-0.74	65191.00
234	3	500	20	2733884	-0.82	64926.00
235	3	500	20	2716049	-1.07	69913.00
236	3	500	20	2721673	-0.88	67451.67
237	3	500	20	2708126	-0.95	65053.33

557	6	100	10	86902	-0.37	83.00
558	6	100	10	90251	-0.15	79.33
559	6	100	10	93479	-0.22	84.00
560	6	100	10	91946	0.35	86.00
561	6	100	20	147223	0.26	152.33
562	6	100	20	149304	-0.23	101.67
563	6	100	20	148338	0.13	161.67
564	6	100	20	148962	0.09	164.00
565	6	100	20	147135	-0.02	131.00
566	6	100	20	148348	-0.01	139.33
567	6	100	20	148497	-0.07	99.33
568	6	100	20	153754	-0.12	112.00
569	6	100	20	149515	-0.16	164.00
570	6	100	20	150470	-0.14	159.33
571	6	200	10	265382	-0.05	536.33
572	6	200	10	261730	0.03	546.00
573	6	200	10	264051	-0.44	558.67
574	6	200	10	258771	-0.26	583.33
575	6	200	10	262667	-0.16	547.67
576	6	200	10	258214	0.10	557.00
577	6	200	10	269073	-0.41	533.67
578	6	200	10	265313	0.28	958.33
579	6	200	10	260177	0.16	696.67
580	6	200	10	262392	-0.28	940.67
581	6	200	20	386304	-0.78	1724.00
582	6	200	20	395777	-0.19	1110.33
583	6	200	20	401415	-0.31	1831.00
584	6	200	20	392901	0.00	1606.00
585	6	200	20	388083	-0.12	1129.33
586	6	200	20	388602	-0.36	1117.00
587	6	200	20	394185	-0.39	1127.00
588	6	200	20	394456	-0.26	1076.33
589	6	200	20	394352	-0.13	1876.00
590	6	200	20	395508	-0.36	1095.33
591	6	500	20	1642018	-0.86	36099.33
592	6	500	20	1675339	-0.89	36532.67
593	6	500	20	1656223	-0.70	32643.33
594	6	500	20	1661063	-0.82	38290.00
595	6	500	20	1653795	-0.59	37365.33
596	6	500	20	1656975	-0.57	30457.67
597	6	500	20	1646253	-0.64	32667.33

238	3	500	20	2731185	-0.98	90042.33
239	3	500	20	2709760	-0.78	86144.33
240	3	500	20	2721814	-1.04	85299.33
241	4	20	5	6862	-0.01	0.33
242	4	20	5	7273	-0.02	0.33
243	4	20	5	6405	0.00	0.33
244	4	20	5	7648	0.00	0.33
245	4	20	5	6628	0.00	0.33
246	4	20	5	6469	0.00	0.33
247	4	20	5	6507	-0.06	0.33
248	4	20	5	6837	0.00	0.33
249	4	20	5	7154	0.00	0.33
250	4	20	5	6336	-0.01	0.33
251	4	20	10	12273	0.00	0.67
252	4	20	10	13371	-0.01	0.67
253	4	20	10	11829	-0.01	0.67
254	4	20	10	11039	0.00	0.67
255	4	20	10	11221	0.00	0.67
256	4	20	10	11110	0.00	0.67
257	4	20	10	10922	0.00	0.67
258	4	20	10	11921	0.00	0.67
259	4	20	10	11819	0.00	0.67
260	4	20	10	12584	0.00	0.67
261	4	20	20	22949	0.00	1.00
262	4	20	20	21610	0.00	1.33
263	4	20	20	23144	0.00	1.33
264	4	20	20	21919	0.00	1.33
265	4	20	20	23385	0.00	1.33
266	4	20	20	22187	0.00	1.33
267	4	20	20	22780	0.00	1.33
268	4	20	20	22097	0.00	1.33
269	4	20	20	22898	0.00	1.33
270	4	20	20	21844	-0.01	1.00
271	4	50	5	24075	-0.55	4.33
272	4	50	5	25418	0.04	4.67
273	4	50	5	23562	-0.62	4.67
274	4	50	5	25786	0.16	4.67
275	4	50	5	26099	0.07	4.33
276	4	50	5	25144	0.54	4.67
277	4	50	5	24937	-0.09	4.67
278	4	50	5	24322	-0.04	4.33
598	6	500	20	1663966	-0.52	35442.00
599	6	500	20	1650691	-0.43	37425.00
600	6	500	20	1657206	-0.55	35332.33
601	7	20	5	5869	0.00	0.33
602	7	20	5	6081	0.00	0.33
603	7	20	5	5369	0.00	0.00
604	7	20	5	6497	0.00	0.00
605	7	20	5	5681	0.00	0.00
606	7	20	5	5611	0.00	0.00
607	7	20	5	5513	0.00	0.33
608	7	20	5	5891	0.00	0.00
609	7	20	5	6045	0.00	0.33
610	7	20	5	5387	0.00	0.00
611	7	20	10	11054	0.00	0.33
612	7	20	10	11885	0.00	0.33
613	7	20	10	10633	0.00	0.33
614	7	20	10	9796	0.00	0.33
615	7	20	10	10065	0.00	0.33
616	7	20	10	9972	0.00	0.33
617	7	20	10	10033	0.00	0.33
618	7	20	10	10675	0.00	0.33
619	7	20	10	10677	0.00	0.33
620	7	20	10	11345	0.00	0.33
621	7	20	20	21410	0.00	0.67
622	7	20	20	19987	0.00	1.00
623	7	20	20	21514	0.00	0.67
624	7	20	20	20437	0.00	1.00
625	7	20	20	21672	0.00	0.67
626	7	20	20	20681	0.00	1.00
627	7	20	20	21106	0.00	1.00
628	7	20	20	20595	0.00	1.00
629	7	20	20	21237	0.01	0.67
630	7	20	20	20379	0.00	1.00
631	7	50	5	18192	0.14	3.00
632	7	50	5	19320	0.21	3.00
633	7	50	5	17991	-0.16	3.33
634	7	50	5	19424	-0.04	3.00
635	7	50	5	19523	0.19	3.00
636	7	50	5	19124	0.11	3.33
637	7	50	5	18806	-0.14	3.00
638	7	50	5	18498	0.22	3.00

279	4	50	5	23678	-0.01	4.67
280	4	50	5	25730	-0.38	4.33
281	4	50	10	40193	0.00	9.33
282	4	50	10	38244	-0.34	9.33
283	4	50	10	37262	0.19	9.33
284	4	50	10	39974	0.04	9.00
285	4	50	10	40096	-0.41	9.33
286	4	50	10	39902	-0.30	8.67
287	4	50	10	41040	0.24	8.67
288	4	50	10	39878	-0.07	9.33
289	4	50	10	39773	-0.25	9.33
290	4	50	10	41223	0.34	9.00
291	4	50	20	70521	0.04	17.67
292	4	50	20	66290	-0.04	17.67
293	4	50	20	65574	0.03	17.33
294	4	50	20	67319	-0.02	17.67
295	4	50	20	66474	-0.25	16.00
296	4	50	20	67108	0.05	17.33
297	4	50	20	68336	-0.01	17.67
298	4	50	20	68323	-0.04	17.67
299	4	50	20	68009	-0.18	17.67
300	4	50	20	69075	-0.16	17.67
301	4	100	5	82156	-0.14	42.00
302	4	100	5	78527	-0.32	37.67
303	4	100	5	76668	-0.22	37.00
304	4	100	5	73920	-0.16	36.67
305	4	100	5	77936	0.22	37.00
306	4	100	5	75865	0.39	38.67
307	4	100	5	77416	0.14	37.33
308	4	100	5	74989	-0.09	38.00
309	4	100	5	80832	-0.11	43.67
310	4	100	5	78652	-0.26	40.00
311	4	100	10	115312	0.07	82.00
312	4	100	10	105203	-0.32	84.33
313	4	100	10	111917	0.18	77.33
314	4	100	10	115717	-0.55	82.00
315	4	100	10	110395	-0.10	79.67
316	4	100	10	104692	-0.12	80.67
317	4	100	10	108038	-0.05	75.67
318	4	100	10	112113	0.05	84.00
319	4	100	10	116158	-0.09	80.67

639	7	50	5	17895	0.10	3.00
640	7	50	5	19385	-0.01	3.00
641	7	50	10	32926	0.02	6.00
642	7	50	10	31410	-0.02	6.00
643	7	50	10	30699	-0.16	5.67
644	7	50	10	32820	-0.01	6.00
645	7	50	10	32812	-0.09	6.00
646	7	50	10	32743	-0.15	5.67
647	7	50	10	33271	-0.12	5.67
648	7	50	10	32670	0.00	6.33
649	7	50	10	32530	-0.14	6.00
650	7	50	10	33567	-0.03	5.67
651	7	50	20	61492	-0.05	11.33
652	7	50	20	57791	0.03	11.33
653	7	50	20	56727	-0.08	11.33
654	7	50	20	58323	-0.07	10.67
655	7	50	20	57770	-0.02	11.00
656	7	50	20	58266	-0.03	11.00
657	7	50	20	59407	-0.15	10.67
658	7	50	20	59199	-0.11	10.00
659	7	50	20	59015	0.03	11.00
660	7	50	20	60171	0.15	10.33
661	7	100	5	56581	0.00	26.33
662	7	100	5	53968	-0.58	24.00
663	7	100	5	52719	-0.29	24.00
664	7	100	5	51520	0.47	23.33
665	7	100	5	53502	-0.38	25.67
666	7	100	5	52245	0.20	26.00
667	7	100	5	53406	0.08	23.67
668	7	100	5	51539	-0.41	26.00
669	7	100	5	55988	0.20	26.33
670	7	100	5	54267	-0.24	26.00
671	7	100	10	86740	0.19	50.33
672	7	100	10	79489	0.05	50.67
673	7	100	10	84154	0.07	54.33
674	7	100	10	87781	0.39	48.00
675	7	100	10	83574	0.05	53.00
676	7	100	10	78869	-0.13	49.00
677	7	100	10	81215	0.04	51.00
678	7	100	10	84189	0.14	48.33
679	7	100	10	87267	0.07	50.33

320	4	100	10	113173	-0.30	79.33
321	4	100	20	171958	-0.08	160.67
322	4	100	20	174487	-0.27	160.33
323	4	100	20	172704	-0.56	160.00
324	4	100	20	174061	-0.22	154.33
325	4	100	20	172392	0.15	162.00
326	4	100	20	173077	-0.41	169.67
327	4	100	20	173884	-0.40	164.33
328	4	100	20	180564	-0.01	163.67
329	4	100	20	174771	-0.36	159.33
330	4	100	20	176650	-0.13	160.00
331	4	200	10	347975	0.02	901.00
332	4	200	10	344837	0.04	892.00
333	4	200	10	347553	-0.09	886.33
334	4	200	10	341854	0.37	858.33
335	4	200	10	343135	-0.34	852.67
336	4	200	10	335734	-0.58	879.67
337	4	200	10	353027	-0.07	873.00
338	4	200	10	346562	-0.15	835.00
339	4	200	10	341798	0.07	810.00
340	4	200	10	344619	0.03	867.67
341	4	200	20	480232	-0.13	1779.00
342	4	200	20	489848	-0.01	1806.00
343	4	200	20	497547	-0.06	1782.00
344	4	200	20	485003	-0.46	1706.00
345	4	200	20	478953	-0.30	1724.33
346	4	200	20	478937	-0.65	1772.33
347	4	200	20	487144	-0.44	1722.67
348	4	200	20	488009	-0.24	1778.67
349	4	200	20	486306	-0.33	1734.33
350	4	200	20	488316	-0.57	1758.33
351	4	500	20	2185976	-0.65	47316.00
352	4	500	20	2219867	-0.93	48685.67
353	4	500	20	2191965	-0.79	48320.00
354	4	500	20	2206167	-0.71	47265.00
355	4	500	20	2191596	-0.79	47874.00
356	4	500	20	2191719	-0.72	50388.33
357	4	500	20	2180549	-0.94	47783.67
358	4	500	20	2203707	-0.85	46726.00
359	4	500	20	2182312	-0.74	47887.00
360	4	500	20	2204013	-0.37	46743.00

680	7	100	10	85679	0.26	48.33
681	7	100	20	139626	0.07	93.33
682	7	100	20	142039	-0.03	91.33
683	7	100	20	140317	-0.22	95.00
684	7	100	20	140992	-0.01	96.00
685	7	100	20	139830	0.13	99.33
686	7	100	20	141109	0.21	96.67
687	7	100	20	141281	0.22	96.33
688	7	100	20	146007	-0.10	98.33
689	7	100	20	142270	0.08	95.00
690	7	100	20	142843	-0.16	93.33
691	7	200	10	241334	-0.02	533.00
692	7	200	10	237157	-0.50	560.33
693	7	200	10	241362	-0.09	525.33
694	7	200	10	235874	-0.17	541.33
695	7	200	10	239161	-0.12	523.00
696	7	200	10	234830	-0.09	539.00
697	7	200	10	246989	0.45	518.00
698	7	200	10	239382	-0.66	514.33
699	7	200	10	235454	-0.49	537.33
700	7	200	10	239328	-0.21	546.33
701	7	200	20	360747	-0.37	1040.67
702	7	200	20	368723	-0.10	1006.67
703	7	200	20	373552	-0.27	1065.67
704	7	200	20	364006	-0.42	985.33
705	7	200	20	360965	0.01	1016.33
706	7	200	20	362168	-0.03	981.33
707	7	200	20	367960	-0.01	1034.33
708	7	200	20	367666	-0.09	1040.00
709	7	200	20	366698	0.04	1024.67
710	7	200	20	370805	0.43	1057.00
711	7	500	20	1490925	-0.56	25868.33
712	7	500	20	1520943	-0.58	26214.00
713	7	500	20	1503697	-0.23	27438.67
714	7	500	20	1505145	-0.73	26775.67
715	7	500	20	1501044	-0.38	26882.67
716	7	500	20	1502593	-0.24	25644.00
717	7	500	20	1495647	-0.37	26050.67
718	7	500	20	1505397	-0.44	28232.00
719	7	500	20	1485826	-1.03	26835.67
720	7	500	20	1499961	-0.61	27231.67

Appendix B: Results of no-wait homogenous DPFSP problems in chapter 5

Problems				Results		
No	f	n	m	Makespan	RPD	CPU (S)
1	2	20	5	3407	-4.49	11
2	2	20	5	3563	-1.33	13
3	2	20	5	3119	-7.67	12
4	2	20	5	3509	-3.15	13
5	2	20	5	3180	-0.22	12
6	2	20	5	3386	-2.05	12
7	2	20	5	3139	-7.59	11
8	2	20	5	3328	-6.88	11
9	2	20	5	3429	-3.52	13
10	2	20	5	3143	-2.45	13
11	2	20	10	4398	-4.16	27
12	2	20	10	4668	-3.45	19
13	2	20	10	4050	-5.46	22
14	2	20	10	3727	-5.33	19
15	2	20	10	4302	-3.56	16
16	2	20	10	3997	0.00	16
17	2	20	10	3963	-6.16	19
18	2	20	10	4228	-7.22	27
19	2	20	10	4281	-5.16	19
20	2	20	10	4429	-7.07	16
21	2	20	20	6038	-4.96	35
22	2	20	20	5872	-3.90	31
23	2	20	20	6502	-0.20	40
24	2	20	20	6261	-3.93	30
25	2	20	20	6156	0.00	30
26	2	20	20	6117	-2.50	42
27	2	20	20	6125	-6.26	31
28	2	20	20	5752	-1.66	30
29	2	20	20	5732	-9.40	44
30	2	20	20	6070	-3.70	35
31	2	50	5	7169	-2.44	162
32	2	50	5	7535	-2.99	147
33	2	50	5	7021	-2.42	141
34	2	50	5	7476	-4.36	133
35	2	50	5	7450	-4.25	140

Problems				Results		
No	f	n	m	Makespan	RPD	CPU (S)
331	5	20	5	1628	-3.55	5
332	5	20	5	1541	-2.34	5
333	5	20	5	1266	-10.15	5
334	5	20	5	1717	-5.45	5
335	5	20	5	1570	-4.56	6
336	5	20	5	1535	-15.15	6
337	5	20	5	1397	-8.81	6
338	5	20	5	1527	-7.23	7
339	5	20	5	1484	-8.90	4
340	5	20	5	1401	-1.62	6
341	5	20	10	2328	-6.36	13
342	5	20	10	2507	-8.54	8
343	5	20	10	2108	-9.61	9
344	5	20	10	1929	-9.01	11
345	5	20	10	2191	-12.08	13
346	5	20	10	2102	-5.66	9
347	5	20	10	2154	-5.90	9
348	5	20	10	2212	-5.15	12
349	5	20	10	2245	-10.16	8
350	5	20	10	2508	-3.83	8
351	5	20	20	4112	-5.56	17
352	5	20	20	3791	-3.73	27
353	5	20	20	4027	-4.03	17
354	5	20	20	3967	-7.01	27
355	5	20	20	4265	-1.04	16
356	5	20	20	4035	-3.75	20
357	5	20	20	4098	-5.45	14
358	5	20	20	4024	-4.73	20
359	5	20	20	4016	-6.60	18
360	5	20	20	3993	-3.08	20
361	5	50	5	2742	-0.54	48
362	5	50	5	2778	-4.63	45
363	5	50	5	2662	-8.30	48
364	5	50	5	2820	-2.52	44
365	5	50	5	2896	-7.33	43

36	2	50	5	7540	-0.97	147	366	5	50	5	2840	-1.97	37
37	2	50	5	7080	-2.33	150	367	5	50	5	2806	-2.33	40
38	2	50	5	7142	-4.40	150	368	5	50	5	2635	-8.25	61
39	2	50	5	7020	-1.90	152	369	5	50	5	2431	-8.75	44
40	2	50	5	7262	-2.94	143	370	5	50	5	2699	-10.39	42
41	2	50	10	9044	-3.29	237	371	5	50	10	3809	-3.42	73
42	2	50	10	8620	-1.23	240	372	5	50	10	3756	-4.60	72
43	2	50	10	8846	-2.15	224	373	5	50	10	3602	-9.54	116
44	2	50	10	8767	-0.43	252	374	5	50	10	3720	-9.09	65
45	2	50	10	9250	-1.59	238	375	5	50	10	3924	-2.78	81
46	2	50	10	9087	-2.28	233	376	5	50	10	3987	-2.99	82
47	2	50	10	9264	-1.66	206	377	5	50	10	3872	-9.60	73
48	2	50	10	8919	-3.97	212	378	5	50	10	3901	-4.18	68
49	2	50	10	8656	-1.75	230	379	5	50	10	3758	-4.69	72
50	2	50	10	9029	-0.75	239	380	5	50	10	3865	-6.35	61
51	2	50	20	11881	-3.25	438	381	5	50	20	6270	-3.03	137
52	2	50	20	11043	-2.82	415	382	5	50	20	5904	-5.37	148
53	2	50	20	11311	-2.54	411	383	5	50	20	6034	-2.74	148
54	2	50	20	11466	-1.49	462	384	5	50	20	6019	-3.22	148
55	2	50	20	11457	-2.83	438	385	5	50	20	6123	-1.50	148
56	2	50	20	11716	-2.51	432	386	5	50	20	5923	-0.97	135
57	2	50	20	11419	-3.02	428	387	5	50	20	5776	-7.67	149
58	2	50	20	11477	-4.95	405	388	5	50	20	6249	-1.51	147
59	2	50	20	11437	-0.26	408	389	5	50	20	6006	-8.54	162
60	2	50	20	11953	-3.07	458	390	5	50	20	6030	-5.60	180
61	2	100	5	14523	-2.67	1138	391	5	100	5	5101	-7.32	232
62	2	100	5	13497	-3.00	1116	392	5	100	5	4906	-3.65	240
63	2	100	5	13396	-2.24	1145	393	5	100	5	4755	-3.65	243
64	2	100	5	13542	-2.36	1075	394	5	100	5	4671	-3.81	252
65	2	100	5	14060	-1.90	1122	395	5	100	5	4850	-2.98	243
66	2	100	5	13357	-3.98	1108	396	5	100	5	4810	-2.39	229
67	2	100	5	13538	-2.79	1110	397	5	100	5	5094	-4.57	265
68	2	100	5	13451	-2.60	996	398	5	100	5	4678	-4.24	209
69	2	100	5	13955	-3.73	1067	399	5	100	5	5075	-1.32	236
70	2	100	5	14196	-1.48	1160	400	5	100	5	5026	-3.64	245
71	2	100	10	16834	-1.59	1796	401	5	100	10	6540	-4.72	414
72	2	100	10	16255	-3.15	1684	402	5	100	10	6611	-3.78	444
73	2	100	10	16673	-2.55	1747	403	5	100	10	6711	-2.43	469
74	2	100	10	16960	-3.54	1811	404	5	100	10	7058	-2.74	411

75	2	100	10	16391	-1.12	1637	405	5	100	10	6558	-2.53	431
76	2	100	10	15983	-3.52	1597	406	5	100	10	6248	-4.01	388
77	2	100	10	15819	-2.36	1724	407	5	100	10	6232	-6.94	438
78	2	100	10	16772	-1.36	1719	408	5	100	10	6684	-4.05	388
79	2	100	10	16644	-1.90	1636	409	5	100	10	6771	-3.63	438
80	2	100	10	17102	-1.51	1629	410	5	100	10	6682	-3.00	367
81	2	100	20	19980	-3.31	3087	411	5	100	20	9347	-2.54	861
82	2	100	20	20901	-1.57	3069	412	5	100	20	9519	-1.15	867
83	2	100	20	20552	-2.14	3176	413	5	100	20	9340	-5.69	813
84	2	100	20	19947	-1.77	3294	414	5	100	20	9223	-2.00	921
85	2	100	20	20402	-1.50	3045	415	5	100	20	9467	-1.36	866
86	2	100	20	20185	-1.62	3351	416	5	100	20	9524	-2.80	1034
87	2	100	20	21138	-1.67	3200	417	5	100	20	9533	-3.58	858
88	2	100	20	20721	-1.09	3108	418	5	100	20	9584	-5.14	853
89	2	100	20	20494	-1.32	3089	419	5	100	20	9480	-3.39	802
90	2	100	20	20759	-2.00	3161	420	5	100	20	9634	-3.18	970
91	2	200	10	31653	-1.32	12805	421	5	200	10	12123	-3.98	2836
92	2	200	10	31166	-2.72	12958	422	5	200	10	11955	-3.46	2541
93	2	200	10	31492	-1.85	13602	423	5	200	10	12321	-3.24	2929
94	2	200	10	30436	-2.45	12506	424	5	200	10	11672	-2.24	2559
95	2	200	10	30784	-2.22	12823	425	5	200	10	11813	-4.05	3028
96	2	200	10	30821	-2.73	13081	426	5	200	10	11471	-5.00	2748
97	2	200	10	31971	-2.47	12876	427	5	200	10	11951	-3.42	2644
98	2	200	10	32225	-1.80	13279	428	5	200	10	12403	-2.94	2819
99	2	200	10	30922	-2.28	12874	429	5	200	10	11692	-2.70	2542
100	2	200	10	32024	-1.76	13458	430	5	200	10	11761	-6.02	4070
101	2	200	20	37820	-1.79	24822	431	5	200	20	16059	-2.61	5860
102	2	200	20	37922	-2.22	23847	432	5	200	20	16311	-3.15	5683
103	2	200	20	38882	-1.66	24902	433	5	200	20	16512	-1.02	5910
104	2	200	20	38562	-1.98	25060	434	5	200	20	16166	-3.04	6277
105	2	200	20	37741	-2.38	24734	435	5	200	20	16177	-2.84	5693
106	2	200	20	38090	-1.30	23945	436	5	200	20	16188	-3.52	6294
107	2	200	20	38220	-1.88	24031	437	5	200	20	16263	-4.22	5659
108	2	200	20	38339	-2.23	25051	438	5	200	20	16465	-2.86	6325
109	2	200	20	37952	-1.42	29120	439	5	200	20	16220	-3.56	5900
110	2	200	20	38203	-1.83	29732	440	5	200	20	16492	-1.85	7421
111	3	20	5	2180	-3.15	5	441	6	20	5	1259	-8.30	4
112	3	20	5	2183	-7.07	10	442	6	20	5	1262	-7.48	4
113	3	20	5	1829	-2.14	5	443	6	20	5	1055	-7.94	4
114	3	20	5	2180	-8.71	6	444	6	20	5	1371	-8.29	4

115	3	20	5	2119	0.00	7
116	3	20	5	2122	-3.68	6
117	3	20	5	1923	-8.03	5
118	3	20	5	2150	-1.47	5
119	3	20	5	2124	-3.63	6
120	3	20	5	1908	-1.14	6
121	3	20	10	3159	-2.32	15
122	3	20	10	3208	-6.77	8
123	3	20	10	2963	-4.48	12
124	3	20	10	2620	-11.49	10
125	3	20	10	2963	-7.32	7
126	3	20	10	2552	-10.52	8
127	3	20	10	2761	-4.96	7
128	3	20	10	2951	-8.30	18
129	3	20	10	2856	-4.83	8
130	3	20	10	3167	-1.25	8
131	3	20	20	5144	-4.78	28
132	3	20	20	4924	0.00	21
133	3	20	20	5245	-3.58	20
134	3	20	20	5021	-3.35	21
135	3	20	20	5379	-4.12	21
136	3	20	20	5109	-0.99	21
137	3	20	20	5102	-5.04	18
138	3	20	20	4991	-2.52	21
139	3	20	20	4939	-4.96	33
140	3	20	20	5043	-7.18	18
141	3	50	5	4177	-6.87	47
142	3	50	5	4576	-2.45	56
143	3	50	5	3942	-5.63	63
144	3	50	5	4390	-0.81	57
145	3	50	5	4487	-1.99	57
146	3	50	5	4295	-5.94	104
147	3	50	5	4372	-4.10	67
148	3	50	5	4352	-0.16	62
149	3	50	5	4011	-0.32	61
150	3	50	5	4130	-1.43	56
151	3	50	10	5951	-2.43	109
152	3	50	10	5663	-4.90	99
153	3	50	10	5568	-3.82	106
154	3	50	10	5890	-5.43	98
155	3	50	10	5981	-0.86	99

445	6	20	5	1207	-20.12	5
446	6	20	5	1244	-6.68	5
447	6	20	5	1105	-4.58	4
448	6	20	5	1197	-7.28	4
449	6	20	5	1204	-15.80	5
450	6	20	5	1073	-22.75	5
451	6	20	10	2205	-9.59	8
452	6	20	10	2273	-10.12	11
453	6	20	10	2048	-7.04	8
454	6	20	10	1840	-4.91	9
455	6	20	10	2031	-18.50	8
456	6	20	10	1875	-12.75	7
457	6	20	10	2049	-9.62	8
458	6	20	10	2131	-13.20	11
459	6	20	10	2138	-15.13	7
460	6	20	10	2294	-7.65	7
461	6	20	20	3736	-9.36	15
462	6	20	20	3666	-8.56	15
463	6	20	20	3956	0.00	14
464	6	20	20	3998	-0.94	12
465	6	20	20	3968	-6.15	23
466	6	20	20	3756	-6.50	17
467	6	20	20	4044	-3.81	12
468	6	20	20	3731	-8.78	13
469	6	20	20	3666	-13.05	20
470	6	20	20	3715	-7.17	15
471	6	50	5	2081	-4.50	35
472	6	50	5	2281	-2.56	33
473	6	50	5	1915	-17.24	34
474	6	50	5	2245	-1.84	27
475	6	50	5	2041	-7.48	30
476	6	50	5	2264	-7.18	34
477	6	50	5	2075	-6.99	40
478	6	50	5	2084	-11.17	35
479	6	50	5	1976	-7.10	35
480	6	50	5	2188	-7.64	33
481	6	50	10	3353	-3.95	56
482	6	50	10	3521	0.00	52
483	6	50	10	3312	-9.75	59
484	6	50	10	3443	-11.99	54
485	6	50	10	3348	-7.08	70

156	3	50	10	5726	-4.73	99
157	3	50	10	5917	-5.09	107
158	3	50	10	6000	-5.96	107
159	3	50	10	5669	-2.78	98
160	3	50	10	5984	-2.38	111
161	3	50	20	9518	-2.67	239
162	3	50	20	8720	-0.89	223
163	3	50	20	8871	-3.63	241
164	3	50	20	8639	-4.62	262
165	3	50	20	8521	-4.25	239
166	3	50	20	8987	-1.57	221
167	3	50	20	8617	-6.38	261
168	3	50	20	8909	-0.12	242
169	3	50	20	9006	-3.85	216
170	3	50	20	9155	-1.80	220
171	3	100	5	8445	-3.47	447
172	3	100	5	7958	-1.75	433
173	3	100	5	7672	-1.77	402
174	3	100	5	7716	-1.28	400
175	3	100	5	8009	-2.72	717
176	3	100	5	8078	-3.79	394
177	3	100	5	7957	-2.70	432
178	3	100	5	7609	-3.81	452
179	3	100	5	8357	-1.62	418
180	3	100	5	8253	-3.20	396
181	3	100	10	10737	-1.21	693
182	3	100	10	10515	-2.99	728
183	3	100	10	10324	-2.88	694
184	3	100	10	10691	-2.85	695
185	3	100	10	10090	-3.14	729
186	3	100	10	10391	-2.44	702
187	3	100	10	10159	-4.03	763
188	3	100	10	10586	-1.70	741
189	3	100	10	10811	-2.59	652
190	3	100	10	10592	-2.53	709
191	3	100	20	14800	-2.08	1647
192	3	100	20	15193	-0.91	1621
193	3	100	20	15048	-1.42	1620
194	3	100	20	14527	-2.39	1642
195	3	100	20	15044	-1.02	1646
196	3	100	20	15038	-3.12	1699

486	6	50	10	3416	-8.93	57
487	6	50	10	3521	-6.80	58
488	6	50	10	3486	-6.82	53
489	6	50	10	3330	-0.92	68
490	6	50	10	3574	-6.81	57
491	6	50	20	5662	-4.42	114
492	6	50	20	5271	-3.81	142
493	6	50	20	5448	-5.99	115
494	6	50	20	5511	-4.85	116
495	6	50	20	5409	-4.42	96
496	6	50	20	5293	-2.11	117
497	6	50	20	5355	-4.03	116
498	6	50	20	5487	-1.90	105
499	6	50	20	5376	-6.97	118
500	6	50	20	5533	-9.00	116
501	6	100	5	3712	-6.40	137
502	6	100	5	3685	-2.15	166
503	6	100	5	3647	-2.75	180
504	6	100	5	3461	-6.74	136
505	6	100	5	3746	-1.91	175
506	6	100	5	3578	-5.37	264
507	6	100	5	3532	-4.39	137
508	6	100	5	3575	-8.05	157
509	6	100	5	3715	-9.19	137
510	6	100	5	3905	-3.77	163
511	6	100	10	5874	-2.64	286
512	6	100	10	5387	-6.59	290
513	6	100	10	5722	-4.30	284
514	6	100	10	5755	-2.92	301
515	6	100	10	5717	-2.37	342
516	6	100	10	5525	-2.57	283
517	6	100	10	5503	-6.14	290
518	6	100	10	5702	-5.69	287
519	6	100	10	5991	-2.93	322
520	6	100	10	5883	-3.94	264
521	6	100	20	8052	-6.82	928
522	6	100	20	8384	-2.51	589
523	6	100	20	8203	-4.64	591
524	6	100	20	8342	-2.40	635
525	6	100	20	8249	-5.75	557
526	6	100	20	8394	-4.07	676

197	3	100	20	14915	-4.40	2114
198	3	100	20	15121	-1.66	1557
199	3	100	20	14935	-1.85	1628
200	3	100	20	15280	-1.39	1640
201	3	200	10	19651	-2.72	8198
202	3	200	10	19336	-4.44	7481
203	3	200	10	19972	-2.77	7701
204	3	200	10	19393	-2.21	7197
205	3	200	10	19727	-4.66	8090
206	3	200	10	19187	-3.02	8153
207	3	200	10	20211	-2.61	7084
208	3	200	10	19895	-1.50	7838
209	3	200	10	19002	-2.38	7535
210	3	200	10	19542	-2.50	8006
211	3	200	20	26882	-1.30	18818
212	3	200	20	27021	-1.92	17820
213	3	200	20	26610	-3.44	18092
214	3	200	20	27031	-2.05	17404
215	3	200	20	27051	-1.48	19878
216	3	200	20	26833	-2.46	19653
217	3	200	20	27123	-1.44	18090
218	3	200	20	27654	-1.06	17690
219	3	200	20	27231	-2.16	17512
220	3	200	20	27071	-2.37	15334
221	4	20	5	1808	-6.47	5
222	4	20	5	1874	-4.44	6
223	4	20	5	1385	-7.67	6
224	4	20	5	1892	-1.92	5
225	4	20	5	1812	-2.27	7
226	4	20	5	1836	-5.21	7
227	4	20	5	1583	-6.99	6
228	4	20	5	1747	-10.18	6
229	4	20	5	1804	-3.53	5
230	4	20	5	1627	-7.82	6
231	4	20	10	2516	-3.53	12
232	4	20	10	2785	-2.28	9
233	4	20	10	2385	-9.80	10
234	4	20	10	2039	-16.61	14
235	4	20	10	2287	-11.36	14
236	4	20	10	2278	-2.98	9
237	4	20	10	2272	-5.73	8

527	6	100	20	8360	-2.25	629
528	6	100	20	8533	-2.56	591
529	6	100	20	8281	-4.48	555
530	6	100	20	8223	-5.69	598
531	6	200	10	10359	-2.68	1960
532	6	200	10	10181	-3.62	2574
533	6	200	10	10515	-2.62	1822
534	6	200	10	10229	-3.95	1761
535	6	200	10	10135	-3.59	1753
536	6	200	10	10104	-3.89	1714
537	6	200	10	10202	-4.20	1626
538	6	200	10	10299	-6.05	1804
539	6	200	10	10117	-3.80	1652
540	6	200	10	10328	-3.06	2605
541	6	200	20	14070	-2.91	3618
542	6	200	20	13798	-5.52	3651
543	6	200	20	14105	-3.41	3782
544	6	200	20	14206	-1.66	3929
545	6	200	20	13777	-5.36	3576
546	6	200	20	14206	-2.08	3909
547	6	200	20	14143	-4.00	3730
548	6	200	20	14348	-0.93	3777
549	6	200	20	14171	-2.97	3583
550	6	200	20	14317	-1.76	3918
551	7	20	5	1152	-10.77	10
552	7	20	5	1061	-13.10	6
553	7	20	5	1049	0.00	5
554	7	20	5	1218	-11.16	6
555	7	20	5	1063	-29.65	5
556	7	20	5	1028	-22.12	5
557	7	20	5	1030	-2.83	5
558	7	20	5	1090	-7.78	4
559	7	20	5	1030	-14.95	5
560	7	20	5	934	-32.76	5
561	7	20	10	2161	-3.74	10
562	7	20	10	2237	-9.14	8
563	7	20	10	1959	-11.08	8
564	7	20	10	1873	-1.89	12
565	7	20	10	1953	-21.63	9
566	7	20	10	1888	-8.04	8
567	7	20	10	2004	-9.36	8

238	4	20	10	2421	-5.61	8
239	4	20	10	2445	-12.05	9
240	4	20	10	2660	-5.57	9
241	4	20	20	4578	-3.44	22
242	4	20	20	4227	-5.63	19
243	4	20	20	4767	-0.96	19
244	4	20	20	4697	-5.26	19
245	4	20	20	4819	-3.77	19
246	4	20	20	4604	-3.78	26
247	4	20	20	4598	-1.94	19
248	4	20	20	4621	-1.01	23
249	4	20	20	4430	-6.89	16
250	4	20	20	4512	-4.39	22
251	4	50	5	3272	-2.71	55
252	4	50	5	3449	-7.41	49
253	4	50	5	3251	-9.49	51
254	4	50	5	3544	-1.39	50
255	4	50	5	3474	-8.41	46
256	4	50	5	3463	-3.13	61
257	4	50	5	3515	-4.51	68
258	4	50	5	3416	-3.64	56
259	4	50	5	3058	-2.11	43
260	4	50	5	3277	-5.01	76
261	4	50	10	4484	-6.02	86
262	4	50	10	4447	-4.30	93
263	4	50	10	4313	-6.83	77
264	4	50	10	4404	-6.22	94
265	4	50	10	4556	-3.08	85
266	4	50	10	4504	-4.92	79
267	4	50	10	4588	-4.26	95
268	4	50	10	4423	-7.06	78
269	4	50	10	4376	-0.14	86
270	4	50	10	4614	-5.00	78
271	4	50	20	7682	-0.12	196
272	4	50	20	7079	-1.35	196
273	4	50	20	7181	-3.86	216
274	4	50	20	7192	-4.09	161
275	4	50	20	7204	0.00	173
276	4	50	20	7147	-4.83	199
277	4	50	20	7182	-7.54	193
278	4	50	20	7153	-0.14	192

568	7	20	10	1948	-8.16	9
569	7	20	10	2062	-17.52	7
570	7	20	10	2244	-9.66	9
571	7	20	20	3745	-5.00	16
572	7	20	20	3605	-9.67	14
573	7	20	20	3653	-1.03	15
574	7	20	20	3862	-3.55	15
575	7	20	20	4107	-2.86	14
576	7	20	20	3595	-7.56	17
577	7	20	20	3694	-9.99	19
578	7	20	20	3531	-11.01	17
579	7	20	20	3547	-12.70	22
580	7	20	20	3580	-7.92	14
581	7	50	5	1816	-8.84	32
582	7	50	5	1860	-14.40	32
583	7	50	5	1753	-11.15	31
584	7	50	5	1841	-4.96	40
585	7	50	5	1822	-8.49	37
586	7	50	5	1982	-7.38	32
587	7	50	5	1898	-7.50	32
588	7	50	5	1844	-8.17	29
589	7	50	5	1637	-15.05	40
590	7	50	5	1867	-5.85	30
591	7	50	10	3190	-2.06	56
592	7	50	10	3075	-2.63	56
593	7	50	10	3093	-9.75	57
594	7	50	10	3148	-14.90	52
595	7	50	10	3064	-4.04	60
596	7	50	10	3283	-2.78	57
597	7	50	10	3292	-6.64	56
598	7	50	10	3180	-8.83	57
599	7	50	10	3046	-1.07	56
600	7	50	10	3239	-8.14	52
601	7	50	20	5203	-5.96	104
602	7	50	20	4860	-3.30	106
603	7	50	20	5017	-1.10	101
604	7	50	20	5011	-5.58	96
605	7	50	20	4782	-6.09	103
606	7	50	20	4968	-4.48	112
607	7	50	20	4903	-8.24	105
608	7	50	20	4936	-5.26	113

279	4	50	20	7204	-7.51	199
280	4	50	20	7539	-3.48	191
281	4	100	5	6537	-1.68	303
282	4	100	5	6087	-1.93	381
283	4	100	5	5841	-1.80	327
284	4	100	5	6045	-2.75	331
285	4	100	5	6217	-5.36	440
286	4	100	5	6070	-2.86	395
287	4	100	5	6153	-3.59	346
288	4	100	5	5922	-4.73	286
289	4	100	5	6355	-2.37	396
290	4	100	5	6494	-4.43	357
291	4	100	10	8139	-1.33	530
292	4	100	10	7653	-5.18	537
293	4	100	10	7845	-4.06	533
294	4	100	10	8011	-4.54	529
295	4	100	10	7768	-3.62	559
296	4	100	10	7640	-4.30	509
297	4	100	10	7622	-2.66	530
298	4	100	10	7870	-4.62	658
299	4	100	10	8168	-2.56	525
300	4	100	10	8054	-3.48	498
301	4	100	20	11581	-3.55	1154
302	4	100	20	11966	-1.73	1247
303	4	100	20	11765	-2.42	1250
304	4	100	20	11476	-1.71	1325
305	4	100	20	11732	-3.33	1184
306	4	100	20	11883	-2.25	1234
307	4	100	20	11850	-4.64	1183
308	4	100	20	11925	-3.18	1169
309	4	100	20	11722	-3.55	1246
310	4	100	20	12148	-1.32	1311
311	4	200	10	15136	-2.52	4221
312	4	200	10	14682	-2.35	3726
313	4	200	10	14705	-3.43	4019
314	4	200	10	14041	-3.23	3876
315	4	200	10	14688	-1.79	3712
316	4	200	10	14361	-3.98	3688
317	4	200	10	14543	-4.19	3922
318	4	200	10	14984	-3.87	4156
319	4	200	10	14262	-2.29	3702

609	7	50	20	5059	-11.60	114
610	7	50	20	5274	-3.92	103
611	7	100	5	3335	-8.23	135
612	7	100	5	3030	-7.87	224
613	7	100	5	3183	-2.87	150
614	7	100	5	3005	-9.49	132
615	7	100	5	3252	-4.75	209
616	7	100	5	3173	-3.91	127
617	7	100	5	3008	-7.59	128
618	7	100	5	2975	-7.18	217
619	7	100	5	3331	-8.81	160
620	7	100	5	3236	-5.32	257
621	7	100	10	5222	-2.45	245
622	7	100	10	4957	-6.44	258
623	7	100	10	4989	-4.86	261
624	7	100	10	5325	-2.20	262
625	7	100	10	5111	-3.33	274
626	7	100	10	4988	-3.76	244
627	7	100	10	4914	-6.02	305
628	7	100	10	5036	-5.55	260
629	7	100	10	5160	-3.75	274
630	7	100	10	5218	-3.73	261
631	7	100	20	7445	-6.54	577
632	7	100	20	7596	-1.26	539
633	7	100	20	7545	-2.67	550
634	7	100	20	7262	-3.33	538
635	7	100	20	7480	-3.16	507
636	7	100	20	7522	-1.75	543
637	7	100	20	7452	-5.67	542
638	7	100	20	7782	-4.54	535
639	7	100	20	7579	-5.97	511
640	7	100	20	7631	-4.47	541
641	7	200	10	9352	-4.23	1625
642	7	200	10	9029	-2.08	1494
643	7	200	10	9035	-3.77	1706
644	7	200	10	8929	-2.79	1545
645	7	200	10	8887	-4.27	1698
646	7	200	10	8781	-2.87	1610
647	7	200	10	8740	-6.18	1544
648	7	200	10	9162	-2.86	1657
649	7	200	10	8760	-3.48	1439

320	4	200	10	14551	-3.86	5815
321	4	200	20	20732	-1.93	8826
322	4	200	20	20682	-2.50	8068
323	4	200	20	20951	-2.40	9057
324	4	200	20	20528	-4.04	10157
325	4	200	20	20725	-2.47	8954
326	4	200	20	20457	-2.63	8713
327	4	200	20	20883	-2.37	8234
328	4	200	20	20818	-1.85	8679
329	4	200	20	20731	-1.95	9461
330	4	200	20	20741	-3.42	10889

650	7	200	10	8773	-5.01	1488
651	7	200	20	12541	-2.02	3334
652	7	200	20	12306	-3.44	3097
653	7	200	20	12649	-1.69	3081
654	7	200	20	12437	-2.45	3290
655	7	200	20	12286	-3.58	3093
656	7	200	20	12441	-2.28	3176
657	7	200	20	12559	-1.70	3207
658	7	200	20	12280	-5.32	2934
659	7	200	20	12526	-2.96	3205
660	7	200	20	12342	-3.44	3235

Where n = number of jobs, f = number of factories, m = number of machines

Appendix C: Results of no-wait heterogenous DPFSP problems in chapter 5

Problems				Results		
No	f	n	m	Makespan	RPD	CPU
1	2	20	5	836	1.09	4.33
2	2	20	5	899	6.52	3.67
3	2	20	5	800	1.65	3.67
4	2	20	5	887	2.07	3.67
5	2	20	5	838	5.01	4.33
6	2	20	5	849	5.20	3.67
7	2	20	5	834	1.83	3.33
8	2	20	5	844	2.55	3.67
9	2	20	5	886	8.45	3.67
10	2	20	5	770	3.36	3.67
11	2	20	10	1250	2.97	7.00
12	2	20	10	1284	-0.47	8.00
13	2	20	10	1203	7.03	7.00
14	2	20	10	1085	1.78	6.67
15	2	20	10	1186	5.99	7.00
16	2	20	10	1165	6.49	7.00
17	2	20	10	1164	2.19	7.00
18	2	20	10	1224	-0.16	8.00
19	2	20	10	1240	5.35	9.67
20	2	20	10	1310	4.88	6.33
21	2	20	20	1934	1.36	11.33
22	2	20	20	1854	3.58	13.67
23	2	20	20	1991	3.21	10.00
24	2	20	20	1971	6.20	13.33
25	2	20	20	1956	1.24	13.00
26	2	20	20	1983	3.88	13.33
27	2	20	20	1948	0.83	13.00
28	2	20	20	1881	3.35	13.33
29	2	20	20	1983	3.61	13.33
30	2	20	20	1942	2.97	11.33
31	2	50	5	1756	-0.73	45.00
32	2	50	5	1873	-0.11	52.67
33	2	50	5	1766	-0.51	50.00
34	2	50	5	1820	-1.94	48.00
35	2	50	5	1834	-0.65	44.67

Problems				Results		
No	f	n	m	Makespan	RPD	CPU
331	5	20	5	459	4.79	2.00
332	5	20	5	482	5.47	1.33
333	5	20	5	421	2.18	2.00
334	5	20	5	519	8.35	1.33
335	5	20	5	463	3.58	1.67
336	5	20	5	466	4.72	1.67
337	5	20	5	459	0.88	2.00
338	5	20	5	465	3.56	2.00
339	5	20	5	464	4.98	1.67
340	5	20	5	418	4.24	1.33
341	5	20	10	776	3.19	3.67
342	5	20	10	822	3.27	3.00
343	5	20	10	760	6.89	3.00
344	5	20	10	668	1.98	3.00
345	5	20	10	720	4.35	2.67
346	5	20	10	702	6.36	2.67
347	5	20	10	723	2.12	3.33
348	5	20	10	764	4.09	3.00
349	5	20	10	759	4.83	3.33
350	5	20	10	789	2.07	3.00
351	5	20	20	1361	2.72	6.00
352	5	20	20	1299	3.67	5.67
353	5	20	20	1379	1.25	5.33
354	5	20	20	1319	1.31	5.33
355	5	20	20	1357	1.57	4.67
356	5	20	20	1353	3.05	4.67
357	5	20	20	1346	1.58	5.33
358	5	20	20	1304	2.68	6.33
359	5	20	20	1344	1.66	5.33
360	5	20	20	1282	1.42	5.33
361	5	50	5	815	6.40	13.00
362	5	50	5	852	2.53	12.00
363	5	50	5	792	2.72	13.67
364	5	50	5	839	0.84	13.67
365	5	50	5	844	2.93	13.67

36	2	50	5	1821	8.65	47.67
37	2	50	5	1752	-2.18	48.00
38	2	50	5	1785	0.90	47.00
39	2	50	5	1695	1.74	50.67
40	2	50	5	1788	-1.97	46.00
41	2	50	10	2418	-1.43	87.00
42	2	50	10	2415	0.50	87.33
43	2	50	10	2321	-0.60	94.33
44	2	50	10	2459	16.76	84.00
45	2	50	10	2428	0.66	89.33
46	2	50	10	2435	6.19	85.67
47	2	50	10	2477	-1.63	89.00
48	2	50	10	2450	1.45	88.67
49	2	50	10	2424	29.70	87.67
50	2	50	10	2431	-0.12	83.00
51	2	50	20	3625	10.55	154.33
52	2	50	20	3436	10.02	173.67
53	2	50	20	3405	-2.41	163.67
54	2	50	20	3410	-1.22	177.33
55	2	50	20	3471	0.75	176.33
56	2	50	20	3465	40.85	155.00
57	2	50	20	3461	0.03	162.67
58	2	50	20	3536	0.03	165.33
59	2	50	20	3507	2.60	155.00
60	2	50	20	3526	-1.07	154.00
61	2	100	5	3499	-1.19	372.67
62	2	100	5	3369	-1.78	356.67
63	2	100	5	3383	1.17	333.00
64	2	100	5	3266	-0.06	337.33
65	2	100	5	3380	0.09	340.00
66	2	100	5	3301	-2.11	339.67
67	2	100	5	3372	-1.29	346.67
68	2	100	5	3355	-2.24	318.67
69	2	100	5	3480	2.96	336.67
70	2	100	5	3488	-0.77	347.67
71	2	100	10	4431	-0.76	638.00
72	2	100	10	4376	-0.50	673.67
73	2	100	10	4440	1.05	647.33
74	2	100	10	4600	0.26	652.00

366	5	50	5	829	0.48	14.67
367	5	50	5	826	1.23	14.67
368	5	50	5	797	2.71	13.33
369	5	50	5	766	-0.13	16.67
370	5	50	5	816	2.64	15.00
371	5	50	10	1214	1.59	26.67
372	5	50	10	1153	-0.43	24.00
373	5	50	10	1153	-0.17	29.33
374	5	50	10	1227	2.16	24.00
375	5	50	10	1198	2.66	24.67
376	5	50	10	1238	8.98	27.00
377	5	50	10	1235	12.48	27.00
378	5	50	10	1194	20.00	27.00
379	5	50	10	1174	1.29	26.67
380	5	50	10	1208	13.11	24.33
381	5	50	20	1979	14.66	44.67
382	5	50	20	1857	4.27	45.67
383	5	50	20	1909	3.52	45.33
384	5	50	20	1870	0.92	49.33
385	5	50	20	1866	1.14	44.33
386	5	50	20	1885	1.67	45.00
387	5	50	20	1905	3.36	44.00
388	5	50	20	1937	3.42	49.00
389	5	50	20	1921	7.38	41.00
390	5	50	20	1956	3.44	45.00
391	5	100	5	1507	0.67	86.67
392	5	100	5	1461	-0.68	80.00
393	5	100	5	1408	-1.33	84.00
394	5	100	5	1413	2.32	83.67
395	5	100	5	1443	10.24	74.00
396	5	100	5	1437	-0.62	72.67
397	5	100	5	1445	0.14	74.67
398	5	100	5	1419	0.50	74.00
399	5	100	5	1485	0.00	93.00
400	5	100	5	1483	1.09	85.00
401	5	100	10	2011	-1.42	135.67
402	5	100	10	1972	1.18	127.67
403	5	100	10	1941	-0.56	142.33
404	5	100	10	2031	4.91	144.33

75	2	100	10	4440	-0.34	691.67
76	2	100	10	4300	-1.56	645.33
77	2	100	10	4239	-4.53	676.00
78	2	100	10	4336	-2.61	654.67
79	2	100	10	4527	-2.10	654.33
80	2	100	10	4527	-0.94	625.33
81	2	100	20	6094	28.35	1225.67
82	2	100	20	5998	3.54	1211.00
83	2	100	20	5958	-2.38	1183.33
84	2	100	20	5964	0.88	1223.67
85	2	100	20	5938	-0.60	1182.00
86	2	100	20	6027	2.47	1173.67
87	2	100	20	6142	16.44	1267.67
88	2	100	20	6074	2.24	1188.67
89	2	100	20	6003	-2.07	1231.67
90	2	100	20	6056	-1.42	1262.33
91	2	200	10	8268	14.37	5092.33
92	2	200	10	8250	0.54	5092.33
93	2	200	10	8382	-0.04	4986.33
94	2	200	10	8275	12.36	4954.67
95	2	200	10	8243	14.33	5090.67
96	2	200	10	8163	8.16	5083.00
97	2	200	10	8437	-1.38	4815.67
98	2	200	10	8309	0.97	5139.33
99	2	200	10	8235	-0.13	4862.00
100	2	200	10	8335	4.74	5293.00
101	2	200	20	10847	1.60	9250.00
102	2	200	20	11061	97.94	9274.67
103	2	200	20	10970	-1.68	9473.33
104	2	200	20	11010	21.18	9483.33
105	2	200	20	10924	50.47	9246.67
106	2	200	20	10935	25.20	9213.00
107	2	200	20	11113	39.30	11179.33
108	2	200	20	10988	37.21	11438.00
109	2	200	20	10925	55.21	11531.67
110	2	200	20	10965	-2.17	11610.33
111	3	20	5	633	2.93	4.00
112	3	20	5	644	3.04	3.67
113	3	20	5	585	4.28	4.00
114	3	20	5	669	4.37	4.00

405	5	100	10	1980	-0.60	138.00
406	5	100	10	1878	28.45	124.67
407	5	100	10	1959	4.76	151.67
408	5	100	10	1993	0.25	135.00
409	5	100	10	2006	-1.81	141.00
410	5	100	10	2039	4.30	133.67
411	5	100	20	2897	0.87	283.00
412	5	100	20	2895	-0.03	298.67
413	5	100	20	2852	5.01	251.67
414	5	100	20	2886	0.94	250.33
415	5	100	20	2928	7.25	235.00
416	5	100	20	2894	0.45	248.00
417	5	100	20	2863	-1.14	280.67
418	5	100	20	2992	17.66	281.33
419	5	100	20	2881	11.75	256.67
420	5	100	20	2929	-2.01	278.67
421	5	200	10	3589	0.20	966.00
422	5	200	10	3554	3.46	950.33
423	5	200	10	3601	3.78	1035.33
424	5	200	10	3536	37.75	916.67
425	5	200	10	3581	1.39	920.33
426	5	200	10	3509	12.61	1013.00
427	5	200	10	3633	2.57	904.00
428	5	200	10	3611	14.02	963.33
429	5	200	10	3543	0.20	951.00
430	5	200	10	3567	6.86	1001.67
431	5	200	20	4868	43.01	1733.67
432	5	200	20	4989	11.21	1784.00
433	5	200	20	4964	50.52	1638.67
434	5	200	20	4919	0.39	1792.00
435	5	200	20	4888	0.14	1708.33
436	5	200	20	4925	0.51	1860.33
437	5	200	20	4884	15.68	1731.00
438	5	200	20	4893	73.88	1701.67
439	5	200	20	4963	39.61	1722.00
440	5	200	20	4928	11.47	1782.67
441	6	20	5	426	2.65	3.33
442	6	20	5	427	2.89	3.00
443	6	20	5	399	5.28	3.67
444	6	20	5	456	3.64	3.00

115	3	20	5	613	2.85	4.67
116	3	20	5	663	10.87	3.00
117	3	20	5	641	8.28	4.33
118	3	20	5	630	5.35	2.33
119	3	20	5	644	7.33	3.67
120	3	20	5	563	3.87	4.33
121	3	20	10	955	1.06	7.67
122	3	20	10	1049	3.96	5.33
123	3	20	10	910	1.45	9.00
124	3	20	10	862	3.98	7.67
125	3	20	10	911	5.07	6.33
126	3	20	10	905	6.22	6.67
127	3	20	10	930	5.20	9.00
128	3	20	10	959	1.16	6.67
129	3	20	10	943	3.06	8.67
130	3	20	10	960	0.00	6.67
131	3	20	20	1631	4.15	16.33
132	3	20	20	1510	2.30	16.67
133	3	20	20	1664	2.91	6.33
134	3	20	20	1568	2.35	14.00
135	3	20	20	1665	4.72	14.00
136	3	20	20	1616	3.32	16.67
137	3	20	20	1612	2.81	10.00
138	3	20	20	1553	3.53	14.33
139	3	20	20	1627	3.43	14.00
140	3	20	20	1527	0.66	12.00
141	3	50	5	1233	3.27	41.67
142	3	50	5	1317	4.77	41.33
143	3	50	5	1232	2.50	41.00
144	3	50	5	1277	3.40	39.67
145	3	50	5	1300	33.61	42.33
146	3	50	5	1263	8.51	45.33
147	3	50	5	1256	1.95	42.00
148	3	50	5	1219	1.08	42.67
149	3	50	5	1200	16.05	42.33
150	3	50	5	1278	2.65	43.00
151	3	50	10	1756	24.45	76.33
152	3	50	10	1722	3.80	81.33
153	3	50	10	1718	2.75	74.00
154	3	50	10	1791	20.44	79.67
155	3	50	10	1775	2.60	80.00

445	6	20	5	422	1.20	3.33
446	6	20	5	427	4.15	3.00
447	6	20	5	430	0.00	3.33
448	6	20	5	415	0.24	3.33
449	6	20	5	415	1.97	2.67
450	6	20	5	378	3.56	3.00
451	6	20	10	714	2.59	5.33
452	6	20	10	771	2.80	5.67
453	6	20	10	681	0.44	5.00
454	6	20	10	629	2.28	5.67
455	6	20	10	667	0.91	5.00
456	6	20	10	650	4.33	5.00
457	6	20	10	692	3.13	5.33
458	6	20	10	707	1.00	5.00
459	6	20	10	718	2.28	5.67
460	6	20	10	757	4.13	5.00
461	6	20	20	1296	1.73	10.00
462	6	20	20	1226	2.00	10.33
463	6	20	20	1326	0.45	8.67
464	6	20	20	1287	1.10	8.67
465	6	20	20	1349	3.77	7.67
466	6	20	20	1298	2.69	8.67
467	6	20	20	1286	1.66	8.67
468	6	20	20	1258	1.45	10.00
469	6	20	20	1295	1.97	8.67
470	6	20	20	1260	4.56	8.00
471	6	50	5	685	1.93	23.67
472	6	50	5	729	0.14	22.00
473	6	50	5	688	2.38	25.00
474	6	50	5	716	0.14	18.67
475	6	50	5	729	4.44	22.33
476	6	50	5	727	1.11	24.67
477	6	50	5	719	1.84	22.00
478	6	50	5	709	8.74	20.00
479	6	50	5	670	15.32	20.33
480	6	50	5	715	2.58	22.33
481	6	50	10	1077	1.70	43.33
482	6	50	10	1037	27.55	43.33
483	6	50	10	1064	3.40	35.67
484	6	50	10	1091	1.68	39.00
485	6	50	10	1107	5.53	35.67

156	3	50	10	1767	2.79	79.00
157	3	50	10	1823	6.98	70.33
158	3	50	10	1735	1.64	68.00
159	3	50	10	1737	95.17	84.00
160	3	50	10	1733	0.00	48.00
161	3	50	20	2721	2.18	133.33
162	3	50	20	2566	3.34	158.67
163	3	50	20	2609	2.43	149.33
164	3	50	20	2643	3.93	151.00
165	3	50	20	2627	8.15	141.33
166	3	50	20	2633	4.53	123.67
167	3	50	20	2598	2.73	136.00
168	3	50	20	2658	1.96	148.00
169	3	50	20	2602	12.93	159.00
170	3	50	20	2619	1.75	147.33
171	3	100	5	2421	16.45	244.33
172	3	100	5	2352	-0.25	260.00
173	3	100	5	2277	-0.44	259.67
174	3	100	5	2194	-3.48	352.00
175	3	100	5	2322	8.30	308.67
176	3	100	5	2288	-1.08	285.67
177	3	100	5	2317	0.65	298.33
178	3	100	5	2249	1.26	274.33
179	3	100	5	2372	-0.46	319.33
180	3	100	5	2357	-0.04	277.33
181	3	100	10	3129	0.42	531.33
182	3	100	10	3029	-1.43	527.00
183	3	100	10	3056	37.47	426.00
184	3	100	10	3225	1.83	534.67
185	3	100	10	2998	-2.15	477.00
186	3	100	10	2966	1.92	525.00
187	3	100	10	3038	0.10	460.33
188	3	100	10	3026	-2.32	536.67
189	3	100	10	3185	0.44	544.33
190	3	100	10	3209	3.62	547.00
191	3	100	20	4354	2.11	839.33
192	3	100	20	4264	5.75	869.67
193	3	100	20	4300	21.47	947.33
194	3	100	20	4270	-0.97	807.00
195	3	100	20	4253	-1.87	734.33
196	3	100	20	4273	-0.58	867.33

486	6	50	10	1054	3.33	36.33
487	6	50	10	1120	3.51	34.33
488	6	50	10	1074	1.61	40.00
489	6	50	10	1048	1.26	40.00
490	6	50	10	1075	-0.56	33.00
491	6	50	20	1786	0.28	66.00
492	6	50	20	1682	17.70	72.33
493	6	50	20	1717	1.18	73.00
494	6	50	20	1735	3.77	73.67
495	6	50	20	1695	5.61	66.00
496	6	50	20	1697	0.47	60.00
497	6	50	20	1686	1.93	66.00
498	6	50	20	1721	2.99	65.33
499	6	50	20	1758	3.84	66.00
500	6	50	20	1773	3.20	66.67
501	6	100	5	1276	-1.85	116.33
502	6	100	5	1248	-0.24	130.00
503	6	100	5	1221	1.08	116.67
504	6	100	5	1183	0.77	93.00
505	6	100	5	1240	1.14	105.00
506	6	100	5	1180	-0.59	88.33
507	6	100	5	1214	-1.78	83.00
508	6	100	5	1209	1.43	82.00
509	6	100	5	1266	-0.63	92.00
510	6	100	5	1266	1.85	115.67
511	6	100	10	1730	-1.26	152.67
512	6	100	10	1684	1.32	178.33
513	6	100	10	1705	0.59	185.33
514	6	100	10	1779	8.34	179.33
515	6	100	10	1709	6.95	162.00
516	6	100	10	1676	-0.89	168.00
517	6	100	10	1693	-0.53	169.67
518	6	100	10	1716	-1.66	178.33
519	6	100	10	1774	-1.11	194.67
520	6	100	10	1765	1.09	172.33
521	6	100	20	2541	6.54	349.33
522	6	100	20	2510	23.34	343.67
523	6	100	20	2547	1.51	315.00
524	6	100	20	2528	33.83	333.00
525	6	100	20	2513	-2.26	258.00
526	6	100	20	2534	5.94	304.33

197	3	100	20	4407	0.16	990.67
198	3	100	20	4386	2.40	892.00
199	3	100	20	4319	1.41	987.67
200	3	100	20	4347	7.79	891.67
201	3	200	10	5757	39.13	3886.33
202	3	200	10	5680	-1.13	3797.67
203	3	200	10	5685	6.22	3712.00
204	3	200	10	5689	18.97	3691.67
205	3	200	10	5717	-0.38	3694.33
206	3	200	10	5674	3.07	3687.33
207	3	200	10	5714	-2.29	3537.00
208	3	200	10	5761	0.07	3612.33
209	3	200	10	5641	53.20	3299.67
210	3	200	10	5717	-1.35	3660.33
211	3	200	20	7406	51.76	6357.67
212	3	200	20	7657	35.19	7022.67
213	3	200	20	7574	23.76	7005.00
214	3	200	20	7688	1.60	4834.33
215	3	200	20	7578	28.33	5506.67
216	3	200	20	7540	-3.26	6693.00
217	3	200	20	7709	23.94	6476.33
218	3	200	20	7649	95.98	7084.33
219	3	200	20	7672	5.21	4409.67
220	3	200	20	7699	0.34	4373.00
221	4	20	5	528	4.35	1.33
222	4	20	5	536	3.88	1.67
223	4	20	5	492	4.68	1.67
224	4	20	5	556	3.54	1.67
225	4	20	5	516	1.78	2.00
226	4	20	5	519	3.18	1.67
227	4	20	5	532	8.79	1.67
228	4	20	5	526	4.57	2.00
229	4	20	5	515	2.59	2.00
230	4	20	5	475	6.03	1.67
231	4	20	10	848	3.79	3.33
232	4	20	10	909	4.12	4.00
233	4	20	10	805	3.47	3.33
234	4	20	10	737	4.24	3.33
235	4	20	10	773	2.93	3.33
236	4	20	10	753	3.58	2.67
237	4	20	10	808	5.35	3.67

527	6	100	20	2593	7.64	253.67
528	6	100	20	2636	0.69	262.00
529	6	100	20	2607	5.80	239.67
530	6	100	20	2575	20.27	318.67
531	6	200	10	3039	2.91	1070.67
532	6	200	10	2972	9.51	1173.00
533	6	200	10	2979	12.42	1192.33
534	6	200	10	3034	45.45	966.00
535	6	200	10	3035	59.32	900.00
536	6	200	10	2988	3.36	841.67
537	6	200	10	3029	10.07	894.33
538	6	200	10	3017	-2.01	951.00
539	6	200	10	2991	29.20	960.33
540	6	200	10	3043	7.26	991.00
541	6	200	20	4236	8.31	1823.00
542	6	200	20	4231	2.37	2004.00
543	6	200	20	4262	17.06	1948.00
544	6	200	20	4223	1.96	2112.33
545	6	200	20	4223	-0.98	2100.00
546	6	200	20	4151	33.13	2228.00
547	6	200	20	4232	-1.47	1872.33
548	6	200	20	4223	1.27	1895.00
549	6	200	20	4166	16.14	1967.67
550	6	200	20	4180	28.10	2269.00
551	7	20	5	398	2.58	2.00
552	7	20	5	398	3.11	1.67
553	7	20	5	366	3.39	2.00
554	7	20	5	420	1.45	1.67
555	7	20	5	394	0.25	1.67
556	7	20	5	407	4.09	2.00
557	7	20	5	430	0.00	2.00
558	7	20	5	392	-1.01	2.00
559	7	20	5	404	7.16	1.67
560	7	20	5	357	0.85	1.67
561	7	20	10	722	7.60	3.33
562	7	20	10	724	0.00	3.00
563	7	20	10	675	2.90	2.67
564	7	20	10	595	1.54	3.00
565	7	20	10	661	5.09	2.67
566	7	20	10	619	4.21	3.00
567	7	20	10	671	0.00	3.33

238	4	20	10	823	1.11	3.33
239	4	20	10	805	2.68	4.00
240	4	20	10	880	6.15	3.33
241	4	20	20	1464	4.20	6.00
242	4	20	20	1377	4.32	7.33
243	4	20	20	1490	3.26	6.00
244	4	20	20	1423	2.89	5.00
245	4	20	20	1478	3.79	5.00
246	4	20	20	1468	4.86	7.33
247	4	20	20	1444	2.12	5.00
248	4	20	20	1397	3.64	6.67
249	4	20	20	1468	4.26	6.00
250	4	20	20	1388	3.66	5.00
251	4	50	5	983	5.81	17.67
252	4	50	5	1023	1.39	17.00
253	4	50	5	971	3.19	16.67
254	4	50	5	1006	5.89	16.00
255	4	50	5	1014	3.47	17.67
256	4	50	5	1002	2.35	15.67
257	4	50	5	983	2.50	18.67
258	4	50	5	996	4.84	14.00
259	4	50	5	923	1.76	15.67
260	4	50	5	979	1.66	15.67
261	4	50	10	1429	3.93	29.67
262	4	50	10	1381	2.37	32.00
263	4	50	10	1378	3.45	35.00
264	4	50	10	1450	3.94	32.00
265	4	50	10	1406	2.93	28.33
266	4	50	10	1449	5.15	31.67
267	4	50	10	1463	2.16	31.67
268	4	50	10	1410	12.98	32.33
269	4	50	10	1366	0.15	31.67
270	4	50	10	1427	19.02	29.00
271	4	50	20	2309	4.48	53.67
272	4	50	20	2078	1.81	58.33
273	4	50	20	2173	3.13	59.33
274	4	50	20	2167	2.65	65.00
275	4	50	20	2151	3.46	51.33
276	4	50	20	2158	1.94	48.67
277	4	50	20	2172	4.27	53.00
278	4	50	20	2159	0.79	53.33

568	7	20	10	692	4.85	3.33
569	7	20	10	702	0.00	3.33
570	7	20	10	717	1.27	3.00
571	7	20	20	1270	1.03	5.67
572	7	20	20	1191	0.00	6.33
573	7	20	20	1320	0.00	7.33
574	7	20	20	1251	2.21	5.33
575	7	20	20	1274	0.47	4.33
576	7	20	20	1256	0.00	6.00
577	7	20	20	1253	4.24	4.67
578	7	20	20	1227	0.00	5.33
579	7	20	20	1247	1.22	4.67
580	7	20	20	1202	2.12	5.33
581	7	50	5	602	0.17	11.00
582	7	50	5	645	1.26	11.33
583	7	50	5	617	3.01	11.67
584	7	50	5	655	2.99	11.67
585	7	50	5	630	0.64	10.67
586	7	50	5	644	0.16	12.67
587	7	50	5	653	2.67	11.67
588	7	50	5	628	1.13	11.33
589	7	50	5	609	3.22	12.33
590	7	50	5	621	-0.96	11.67
591	7	50	10	995	4.63	20.00
592	7	50	10	964	4.33	18.33
593	7	50	10	972	4.18	22.33
594	7	50	10	999	2.04	22.33
595	7	50	10	981	2.08	20.00
596	7	50	10	971	0.52	22.00
597	7	50	10	1004	2.34	22.00
598	7	50	10	981	2.40	22.00
599	7	50	10	966	3.32	20.33
600	7	50	10	995	1.74	20.33
601	7	50	20	1687	1.14	36.67
602	7	50	20	1545	0.59	33.67
603	7	50	20	1654	5.15	36.67
604	7	50	20	1580	0.13	40.67
605	7	50	20	1588	2.12	37.00
606	7	50	20	1581	0.83	33.33
607	7	50	20	1583	1.28	36.33
608	7	50	20	1598	1.14	33.67

279	4	50	20	2163	1.50	49.33
280	4	50	20	2226	3.15	53.33
281	4	100	5	1844	-0.27	114.00
282	4	100	5	1824	1.62	120.00
283	4	100	5	1732	-0.12	123.33
284	4	100	5	1705	1.97	118.33
285	4	100	5	1741	10.75	106.67
286	4	100	5	1753	-0.45	100.33
287	4	100	5	1771	1.03	109.67
288	4	100	5	1766	2.67	114.67
289	4	100	5	1811	-0.33	118.67
290	4	100	5	1842	0.66	105.33
291	4	100	10	2437	3.09	171.67
292	4	100	10	2404	2.08	214.00
293	4	100	10	2365	-0.67	196.00
294	4	100	10	2553	3.82	196.67
295	4	100	10	2365	0.00	204.33
296	4	100	10	2298	-2.13	168.00
297	4	100	10	2349	1.03	211.33
298	4	100	10	2412	8.31	174.00
299	4	100	10	2456	-1.48	193.00
300	4	100	10	2443	-0.45	196.00
301	4	100	20	3383	23.69	361.33
302	4	100	20	3440	1.24	364.00
303	4	100	20	3407	1.22	330.67
304	4	100	20	3424	39.58	322.33
305	4	100	20	3429	-0.20	362.00
306	4	100	20	3424	4.61	357.00
307	4	100	20	3440	0.94	388.00
308	4	100	20	3520	69.31	383.00
309	4	100	20	3451	6.38	345.33
310	4	100	20	3506	4.00	362.33
311	4	200	10	4365	27.15	1333.33
312	4	200	10	4343	18.50	1327.67
313	4	200	10	4390	10.55	1293.67
314	4	200	10	4383	4.91	1339.33
315	4	200	10	4321	20.06	1410.00
316	4	200	10	4314	26.92	1377.00
317	4	200	10	4416	-0.74	1394.00
318	4	200	10	4437	31.51	1435.33
319	4	200	10	4314	17.16	1336.33

609	7	50	20	1616	0.81	37.00
610	7	50	20	1634	2.32	40.33
611	7	100	5	1113	-1.77	57.67
612	7	100	5	1079	-2.62	61.00
613	7	100	5	1073	15.87	53.33
614	7	100	5	1053	0.48	63.00
615	7	100	5	1087	2.84	54.33
616	7	100	5	1030	9.57	62.33
617	7	100	5	1087	6.67	61.33
618	7	100	5	1043	0.77	65.33
619	7	100	5	1101	-3.51	67.00
620	7	100	5	1076	-4.01	65.33
621	7	100	10	1540	-0.77	102.67
622	7	100	10	1498	-1.19	109.33
623	7	100	10	1504	0.60	108.67
624	7	100	10	1553	2.31	96.33
625	7	100	10	1499	14.51	102.33
626	7	100	10	1480	-0.40	97.33
627	7	100	10	1512	-2.26	96.33
628	7	100	10	1540	1.12	96.33
629	7	100	10	1567	0.19	109.33
630	7	100	10	1585	1.02	103.00
631	7	100	20	2284	-1.30	212.00
632	7	100	20	2296	0.35	187.33
633	7	100	20	2257	1.94	188.00
634	7	100	20	2302	9.10	188.67
635	7	100	20	2277	0.49	187.33
636	7	100	20	2310	0.83	197.33
637	7	100	20	2279	-1.81	188.33
638	7	100	20	2344	-1.60	200.33
639	7	100	20	2333	9.94	178.67
640	7	100	20	2343	12.48	200.33
641	7	200	10	2671	24.87	658.67
642	7	200	10	2629	2.62	646.00
643	7	200	10	2675	-1.18	599.33
644	7	200	10	2622	18.21	644.33
645	7	200	10	2672	27.48	551.00
646	7	200	10	2611	5.41	656.33
647	7	200	10	2674	3.16	598.33
648	7	200	10	2578	20.58	605.00
649	7	200	10	2630	75.68	573.67

320	4	200	10	4385	3.89	1434.00
321	4	200	20	5849	42.94	2499.00
322	4	200	20	5883	-2.87	2572.00
323	4	200	20	5933	16.81	2437.67
324	4	200	20	5977	14.72	2486.00
325	4	200	20	5833	11.49	2650.33
326	4	200	20	5846	2.54	2627.00
327	4	200	20	6010	0.17	2656.67
328	4	200	20	5993	16.53	2684.00
329	4	200	20	6013	11.27	2649.33
330	4	200	20	6039	12.71	2561.67

650	7	200	10	2626	41.33	650.00
651	7	200	20	3696	10.03	1099.67
652	7	200	20	3747	11.25	1150.00
653	7	200	20	3771	12.10	1010.33
654	7	200	20	3684	9.90	1166.67
655	7	200	20	3704	-1.65	1161.00
656	7	200	20	3652	30.01	1107.67
657	7	200	20	3678	25.61	1149.33
658	7	200	20	3734	7.24	1064.67
659	7	200	20	3736	35.07	1147.00
660	7	200	20	3654	-4.14	1007.33

Where n = number of jobs, f = number of factories, m = number of machines